Deutsches Zentrum
für Luft-
und Raumfahrt e. V.

Institut für Aerodynamik
und Strömungstechnik
Göttingen

Bericht FB-Nr. 2009-23

Verfasser A. Lincke

Titel

Verification and Validation of von Kármán Length Scale for Identification of Turbulent Structures

FE Programm : HighPerFlex
Auftraggeber : DLR
Kostenträger : 22413

Zugänglichkeitsstufe : ■ I öffentlich zugänglich ☐ II int. unbeschränkt ext. beschränkt ☐ III int. beschränkt ext. beschränkt ☐ IV vs

Verteiler: 4 AS-C²A²S²E
8 AE
8 Reserve

Umfang des Berichts: 108 Seiten inkl.
1 Tabellen
33 Literaturstellen
69 Bildern

Quelle: Ort: Göttingen
Abteilung: AS-C²A²S²E
Date: November 2009

Institutsleiter Abteilungsleiter Bearbeiter

Prof. A. Dillmann Dr. D. Schwamborn A. Lincke

Übersicht

Dieser Bericht beschreibt die Validierung des Kármán-Sensors für zwei turbulente Testfälle, Kanalströmung bei $Re_\tau = 395$ und Strömung über eine rückwärts gerichtete Stufe bei $Re_h = 37500$. Die Simulationen wurden mit dem DLR THETA Code berechnet, der auf einem Finite Volumen Verfahren und einem Projektionsschema mit Korrekturformel von Rhie und Chow basiert.

Der Kármán-Sensor wurde 2006 von Menter und Egorov eingeführt, basierend auf der vorherigen Arbeit von Rotta (1968). Der Kármán-Sensor L_{vK} ist gegeben durch $L_{vK} = \kappa \frac{S}{U''}$, mit symmetrischem Deformationstensor S, normiertem Laplace des Geschwinidgkeitsfeldes U'' und der Kármán-Konstante $\kappa = 0.41$.

Das Ziel ist es zu zeigen, dass L_{vK} in Relation mit der Wirbellänge steht. Auf Grund dieser Relation soll dann a posteriori bestimmt werden, ob ein Gitter ausreichend fein ist, um die kleinen Strukturen bei einer LES aufzulösen.

Der Sensor wurde in THETA implementiert und für die Testfälle Kanal und rückwärtsgerichtete Stufe ausgewertet. Es zeigt sich, dass L_{vK} bei feineren Gittern kleinere Werte annimmt. Dem Ansatz folgend, dass L_{vK} mit der Größe der turbulenten Strukturen zusammenhängt, wird der Sensor mit einer charakteristischen Gitterweite $\Delta_{yz} := \sqrt{\Delta y \Delta z}$ verglichen, um eine Aussage über das Gitter zu erhalten. Es zeigt sich, dass der Sensor bei isotroper Gitterverfeinerung benutzt werden kann, um die Gitterqualität a posteriori festzustellen. Bei ausreichend feinem Gitter ist $L_{vK}/\Delta_{yz} > 0.8$, bei zu grobem Gitter $L_{vK}/\Delta_{yz} < 0.6$.

Contents

List of figures

List of tables

1. Introduction

This thesis is the result of a co-operation between the Institute of Aerodynamics and Flow Technology (AS) and and its Center for Computer Applications in AeroSpace Science and Engineering ($C^2A^2S^2E$) of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt) and the Institute for Numerical and Applied Mathematics in Göttingen (Institut für Numerische und Angewandte Mathematik). DLR AS $C^2A^2S^2E$ concerns the development of numerical methods for fluid dynamics applications for transport vehicles (aeroplanes, helicopters, trains) and energy systems (combustion, wind turbines).

Computational fluid dynamics (CFD) plays an important role in industry and engineering. For the production of cars, trains and planes it is important to compute surface pressure, distribution and the wall shear stress in order to be able to construct a shape with minimal friction. There is a huge amount of applications; CFD is even applied to construct functional sport dresses or for animation of fluid dynamics in movies and pictures.

This thesis is concerned with the following questions:

1. How can the motion of a fluid be described?
2. How can the motion of a fluid be computed?
3. How can turbulent flows be modeled?
4. How can the efficiency of computations be improved?
5. How is it possible to assess the quality of a simulation of turbulent flow using a sensor, which is a quantity derived from the flow solution?
6. How can a length scale for the turbulent flow structures be derived from the flow solution?
7. What is the Kármán-sensor?
    - How can the Kármán-sensor be computed?
    - How does the sensor behave in a fluid motion?

- How is it possible to assess the quality of a simulation of turbulent flow using the Kármán-sensor?

In Chapter 2 of this thesis we will introduce the Navier-Stokes equations. These partial differential equations describe the velocity, $\vec{u}$, and pressure, p, of a fluid during a certain time interval and within a certain domain. We will see that these equations are nonlinear and time-dependent. Therefore space and time discretization is necessary in order to compute a solution within a discrete space and at discrete time points. The finite volume method gives a local discretization of the domain into control volumes, such that an approximation at each point of the domain is obtained. We will explain explicitly how the discretized equations are solved. In this thesis we will use the DLR-THETA-code (THETA = turbulent heat release extension of the TAU-code).

The goal of sensors for turbulent structures is to predict the size of the eddies at each point of the domain. With this information it can be determined whether to use a coarse or a fine mesh. By using grid adaptation the mesh is adapted, in such a way that in regions where small eddies exist, a sufficiently fine mesh is used and a coarse mesh is used elsewhere. Consequently, the computations on a fine mesh are spent only where they are necessary. Hence the efficiency of a computation is improved and computational costs can be reduced. Another application of sensors is used in detached eddy simulation (DES). The concept of a DES is to use large-eddy simulation (LES) in regions of separated flow and to use the Reynolds averaged Navier-Stokes equations (RANS) model in attached boundary layers. When to use RANS or LES is determined by the mesh and a sensor derived from flow quantities. Recently, the Kármán-sensor has been used as an alternative switch between LES and RANS in hybrid RANS-LES methods [32].

In Chapter 3 we will introduce the Kármán-sensor which will be studied in this thesis. It was first developed in 1930 by Theodore von Kármán. Later the Kármán-sensor was reviewed by Rotta in the 1960s and by Menter in 2003. The Kármán-sensor is a length scale for eddies in turbulent flows. We will state which numerical methods are necessary in order to compute the sensor. The quality of different numerical methods, like computation with Green-Gauß, Least Square and central differences is analyzed by application to simple test functions.

In this thesis we will consider two test cases:

- channel flow at Reynolds number, $Re = 395$,
- backward-facing step at Reynolds number, $Re = 37500$.

In Chapter 4 we will introduce the first test case, the turbulent channel flow. In turbulent flows the structures of the fluid motion (eddies) become very small, such that a very fine mesh would be necessary to resolve the smallest structures – the consequence thereof is a very time-consuming computation, i.e. it would take a long time to obtain a solution or a large number of CPUs would have to be used. To minimize the cost of a computation, turbulence models have to be used. The RANS model gives an approximation for a time averaged velocity field. The cost of RANS is small in comparison to a direct numerical simulation (DNS) but the results are

not always as reliable. In this thesis we will use LES. In Chapter 4 we will present the concept of LES. The idea of LES is to compute the large eddies and to *model* the effect of the small scales onto the large scales. The costs of LES are higher than those of RANS but the solution should be closer to a DNS. We will introduce the wall-adapting local eddy-viscosity (WALE) model which is the LES model we use for the computation on the channel. We will also show how the wall shear stress, which plays an important role in industry and engineering, can be computed from the solution for velocity. Furthermore, we will see that turbulent flows have an arbitrary character, i.e. that the solution of the velocity at a certain location changes from one instant to another. The solution at one time step does not represent the development over the whole time interval. Thus statistical averaging is necessary in order to give an average solution of the fluid motion and thus of the sensor during the whole time interval.

In Chapter 5 and Chapter 6 we will compute the sensor for both test cases in order to observe how the sensor behaves in turbulent flows. In Chapter 5 the numerical results of the computations of the channel flow are shown. We will study the quality of the LES of the DLR-THETA-code by comparing the results to the solution of the DNS of Moser, et al.[20]. The Kármán-sensor is validated for the channel. We will give first hypotheses how to interpret the sensor and how it could be used to improve the efficiency of an LES. In Chapter 6 we will introduce the test case of a flow over a backward-facing step and will present the first results of the velocity profile and of the Kármán-sensor. The conclusions are summarized in Chapter 7.

2. Numerical methods for the incompressible Navier-Stokes equations

In this chapter we will introduce the governing equations for the incompressible viscous fluid flow, the Navier-Stokes equations and show which methods are used in DLR-THETA-code in order to solve them numerically.

2.1 Incompressible Navier-Stokes equations

The evolution of velocity, $\vec{u} : \Omega \times I \longrightarrow \mathbb{R}^3$ and pressure, $p : \Omega \times I \longrightarrow \mathbb{R}$ in a bounded domain $\Omega \subset \mathbb{R}^3$ and time interval, $I = (0, T]$ of an incompressible fluid is described by the incompressible Navier-Stokes equations:

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) - \nu \Delta \vec{u} + \vec{\nabla} p &= \vec{f}, && (2.1) \\ \vec{\nabla} \cdot \vec{u} &= 0, && (2.2) \end{aligned}$$

where ν denotes the kinematic viscosity and $\vec{f}$ a body force. In this thesis we will set density $\rho \equiv 1$. We will use the tensorial notation where $(\vec{u} \otimes \vec{u})_{ij} = u_i u_j$. Equation (2.1) is the *momentum equation*; (2.2) denotes the *continuity equation* which holds in this form only for incompressible fluids. In componentwise notation we obtain for $\vec{u} = (u, v, w)^T$

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} - \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + \frac{\partial p}{\partial x} &= f_x, \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} - \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) + \frac{\partial p}{\partial y} &= f_y, \\ \frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} - \nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + \frac{\partial p}{\partial z} &= f_z, \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0. \end{aligned}$$

The kinematic viscosity, ν, is assumed to be constant. We prescribe an initial condition, $\vec{u}_0$, at time $t_0 = 0$

$$\vec{u}(\cdot, 0) = \vec{u}_0 \quad \text{in} \quad \Omega, \tag{2.3}$$

a no-slip boundary condition on walls, i. e.

$$\vec{u} = \vec{0} \quad \text{on} \quad \Gamma_W \times [0, T] \tag{2.4}$$

and an inflow and outflow boundary condition

$$\vec{u} = \vec{u}_{in} \quad \text{on} \quad \Gamma_{in} \times [0, T], \tag{2.5}$$

$$(\nu \vec{\nabla} \vec{u} - p\mathbb{I}) \cdot \vec{n} = \vec{0} \quad \text{on} \quad \Gamma_{out} \times [0, T]. \tag{2.6}$$

The Navier-Stokes equations are time-dependent due to the time derivative term and nonlinear due to the convective term $\vec{\nabla} \cdot (\vec{u} \otimes \vec{u})$. Furthermore, note that we have a coupled system containing pressure, p, and velocity, $\vec{u}$.

2.2 Variational formulation of the Navier-Stokes equations

In this section we want to introduce the definition of a *weak* and a *strong* solution of the Navier-Stokes equations. Let $d \in \{2, 3\}$ denote the dimension of $\Omega \subset \mathbb{R}^d$. The pair of functions

$$(\vec{u}, p) \in \left[\mathcal{C}^2(\Omega \times (0, T)) \cap \mathcal{C}(\overline{\Omega} \times [0, T])\right]^d \times \mathcal{C}^1(\Omega \times (0, T)), \tag{2.7}$$

is called *classical* or *strong* solution of (2.1) and (2.2) if these equations are fulfilled at *each* point of $\Omega \times (0, T)$ (see [16]).

Contrary to the concept of a strong solution for the Navier-Stokes equations, we require the weak solution to fulfill the equations in the sense of an integral average over the flow domain. Assuming that the domain Ω fulfills the requirements of Gauß' divergence theorem, we will denote the boundary of Ω with $\partial\Omega$, i.e $\partial\Omega := \overline{\Omega} \setminus \Omega$. For simplicity let us also assume that $\vec{u}(\vec{x}, t) \in \mathcal{C}^1(\overline{\Omega})$ and $\vec{u}(\vec{x}, t) = \vec{0}$ for all $\vec{x} \in \partial\Omega$ and for all $t \in [0, T]$. Let the pair $(\vec{u}, p)$ be a classical solution of (2.1). We choose suitable test functions $(\vec{v}, q) \in \mathcal{C}_0^\infty(\Omega)^d \times \mathcal{C}^\infty(\Omega)$. Therefore $\vec{v}$ vanishes at the boundary of the domain. We multiply the equations (2.1) with $\vec{v}$ and q, respectively. By integration over Ω we obtain a weak formulation of the Navier-Stokes equations which reads:

Find $(\vec{u}, p)$ such that for all $(\vec{v}, q) \in \mathcal{C}_0^\infty(\Omega)^d \times \mathcal{C}^\infty(\Omega)$:

$$\int_\Omega \vec{v} \cdot \frac{\partial \vec{u}}{\partial t} d\Omega \; + \; \int_\Omega \vec{v} \cdot \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) d\Omega \; - \; \int_\Omega \vec{v} \cdot \nu \Delta \vec{u} d\Omega \; + \; \int_\Omega \vec{v} \cdot \vec{\nabla} p d\Omega = \int_\Omega \vec{v} \cdot \vec{f} d\Omega, \tag{2.8}$$

$$\int_\Omega q \vec{\nabla} \cdot \vec{u} d\Omega = 0. \tag{2.9}$$

The standard weak formulation is obtained by additionally intgrating the third and fourth left-hand side terms by parts. This will be done in a similar way in Section 2.4.1. where the special *ansatz* of the test functions of the finite volume method is considered. For the "standard" variational formulation of the Navier-Stokes equations it is proven that a weak solution exists. The pair of functions $(\vec{u}, p)$ which fulfills the variational formulation of (2.1) and(2.2) is called *weak solution*. For further details see [11, p.39].

N.B. 2.2.1. *We will use two different notations for the terms of the Navier-Stokes equations. The convective term, $\vec{\nabla}\cdot(\vec{u}\otimes\vec{u})$, can alternatively be written as $(\vec{u}\cdot\vec{\nabla})\vec{u}$ which is shown in the following calculation. It holds*

$$\vec{\nabla}\cdot(\vec{u}\otimes\vec{u}) \equiv \left(\vec{\nabla}^T\cdot(\vec{u}\otimes\vec{u})\right)^T = \left((\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z})\cdot\begin{pmatrix} u^2 & uv & uw\\ uv & v^2 & vw\\ uw & uv & w^2\end{pmatrix}\right)^T$$
$$= \begin{pmatrix} \frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\frac{\partial uw}{\partial z}\\ \frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}+\frac{\partial vw}{\partial z}\\ \frac{\partial uw}{\partial x}+\frac{\partial vw}{\partial y}+\frac{\partial w^2}{\partial z}\end{pmatrix}$$

and, by using the product rule for differentiation and the continuity equation

$$(\vec{u}\cdot\vec{\nabla})\vec{u} \equiv (\vec{u}^T\cdot\vec{\nabla})\vec{u} = \left((u,v,w)\cdot\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\right)\begin{pmatrix}u\\ v\\ w\end{pmatrix}$$
$$= (u\frac{\partial}{\partial x}+v\frac{\partial}{\partial y}+w\frac{\partial}{\partial z})\cdot\begin{pmatrix}u\\ v\\ w\end{pmatrix} = \begin{pmatrix} u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\\ u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\\ u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\end{pmatrix}$$
$$= \begin{pmatrix} \frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\frac{\partial uw}{\partial z}\\ \frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}+\frac{\partial vw}{\partial z}\\ \frac{\partial uw}{\partial x}+\frac{\partial vw}{\partial y}+\frac{\partial w^2}{\partial z}\end{pmatrix} \underbrace{- \begin{pmatrix} u\left[\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right]\\ v\left[\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right]\\ w\left[\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right]\end{pmatrix}}_{=\vec{u}\vec{\nabla}\cdot\vec{u}=0}$$
$$= \vec{\nabla}\cdot(\vec{u}\otimes\vec{u}).$$

Note that this relation holds if $\vec{u}(\vec{x})$ is sufficiently smooth and if $\vec{\nabla}\cdot\vec{u}=0$.

The diffusive term $\Delta\vec{u}$ can alternatively be written using the symmetric rate-of-strain tensor (see [10])

$$\mathbb{S}(\vec{u}) := \frac{1}{2}\cdot\left(\vec{\nabla}\vec{u}+(\vec{\nabla}\vec{u})^T\right) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)_{i,j}. \tag{2.10}$$

Note that both notations are not equivalent in general. Equivalence can be shown if certain conditions hold which will be derived in the following calculation. Using the symmetric rate-of-strain tensor the Navier-Stokes equations read:

$$\frac{\partial\vec{u}}{\partial t} \;+\; \vec{\nabla}\cdot(\vec{u}\otimes\vec{u}) \;-\; 2\nu\vec{\nabla}\cdot\mathbb{S}(\vec{u}) + \vec{\nabla}p = \vec{f}. \tag{2.11}$$

The variational formulation of (2.11) and (2.2) is obtained by multiplication with suitable test functions $\vec{v}, q$ and by integration over Ω (see section 2.2). We obtain:

$$\int_\Omega \vec{v}\cdot\frac{\partial\vec{u}}{\partial t}d\Omega \;+\; \int_\Omega \vec{v}\cdot\vec{\nabla}\cdot(\vec{u}\otimes\vec{u})d\Omega \;-\; \int_\Omega \vec{v}\cdot\vec{\nabla}\cdot 2\nu\mathbb{S}(\vec{u})d\Omega \;+\; \int_\Omega \vec{v}\cdot\vec{\nabla}p\;d\Omega = \int_\Omega \vec{v}\cdot\vec{f}d\Omega,$$
$$\int_\Omega q\vec{\nabla}\cdot\vec{u}\;d\Omega \;=\; 0.$$

Consequently we have to show that

$$-\int_\Omega \vec{v}\cdot\vec{\nabla}\cdot 2\nu\mathbb{S}(\vec{u})d\Omega \equiv -\int_\Omega \Delta\vec{u}\cdot\vec{v}d\Omega.$$

By using integration by parts for the integral including $\mathbb{S}(\vec{u})$ *we obtain:*

$$\int_\Omega \vec{v}\cdot\frac{\partial\vec{u}}{\partial t}d\Omega \;+\; \int_\Omega \vec{v}\cdot\vec{\nabla}\cdot(\vec{u}\otimes\vec{u})d\Omega \;+\; \int_\Omega \vec{\nabla}\vec{v}:2\nu\mathbb{S}(\vec{u})d\Omega + \int_\Omega \vec{v}\cdot\vec{\nabla}p\;d\Omega = \int_\Omega \vec{v}\cdot\vec{f}d\Omega,$$
$$\int_\Omega q\vec{\nabla}\cdot\vec{u}\;d\Omega \;=\; 0$$

with $\vec{\nabla}\vec{v}:\mathbb{S}(\vec{u}) = \sum_{i,j=1}^3(\vec{\nabla}\vec{v})_{ij}\cdot S(\vec{u})_{ij}$*. The integrand including* $\mathbb{S}(\vec{u})$ *can be transformed to:*

$$\nu\int_\Omega 2\mathbb{S}(\vec{u}):\vec{\nabla}\vec{v}d\Omega = \nu\left(\int_\Omega \vec{\nabla}\vec{u}:\vec{\nabla}\vec{v}d\Omega + \int_\Omega(\vec{\nabla}\vec{u})^T:\vec{\nabla}\vec{v}d\Omega\right). \tag{2.12}$$

Using integration by parts we obtain for the second integral:

$$\begin{aligned}
\int_\Omega(\vec{\nabla}\vec{u})^T:\vec{\nabla}\vec{v}d\Omega &= \int_\Omega\sum_{i,j=1}^3(\frac{\partial u_j}{\partial x_i})(\frac{\partial v_i}{\partial x_j})d\Omega\\
&= -\int_\Omega\sum_{i,j=1}^3 u_j(\frac{\partial^2 v_i}{\partial x_j\partial x_i})d\Omega + \underbrace{\int_{\partial\Omega}\sum_{i,j=1}^3 u_j\frac{\partial v_i}{\partial x_j}\cdot\vec{n}dS}_{=0}\\
&= -\int_\Omega\sum_{i,j=1}^3 u_j(\frac{\partial^2 v_i}{\partial x_j\partial x_i})d\Omega\\
&= \int_\Omega\sum_{i,j=1}^3\frac{\partial u_j}{\partial x_j}\frac{\partial v_i}{\partial x_i}d\Omega - \underbrace{\int_{\partial\Omega}\sum_{i,j=1}^3 u_j\frac{\partial v_i}{\partial x_i}dS}_{=0}\\
&= \int_\Omega\sum_{j=1}^3\frac{\partial u_j}{\partial x_j}\sum_{i=1}^3\frac{\partial v_i}{\partial x_i}d\Omega\\
&= \int_\Omega\underbrace{\vec{\nabla}\cdot\vec{u}}_{=0}\cdot\vec{\nabla}\cdot v d\Omega = 0
\end{aligned}$$

due to the no-slip boundary condition and the continuity equation. By using these transformations for the integral in (2.12) *we obtain by using integration by parts:*

$$\int_\Omega(\vec{\nabla}\vec{v}):2\nu\mathbb{S}(\vec{u})d\Omega = \int_\Omega\nu\vec{\nabla}\vec{u}:\vec{\nabla}\vec{v}d\Omega = -\int_\Omega\Delta\vec{u}\cdot\vec{v}d\Omega. \tag{2.13}$$

As a result both notations are the same in a weak sense under the continuity equation and when the homogenous Dirichlet boundary condition $\vec{u}=\vec{0}$ *on* $\partial\Omega$ *holds.*

2.3 Local discretization on primary and dual grids

Primary and dual grids give a local discretization of the domain Ω. A grid consists of polyedric elements. Each cell of the primary grid is either a tetrahedron, a prism

or a hexahedron. For the sake of simplicity we will, in this section, only consider grids which consist of equally sized rectangulars (cells or faces) in 2D. There is no gap between two cells. The primary grid consists of nodes and edges. The nodes are at the vertices of the edges.

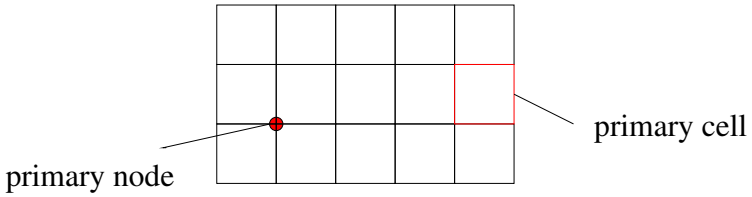

FIGURE 1: Sketch of a primary grid.

The *dual* grid is derived from the primary grid. To each primary node we associate a dual cell. In our case the dual grid consists of rectangulars whose midpoints are the nodes of the primary grid.

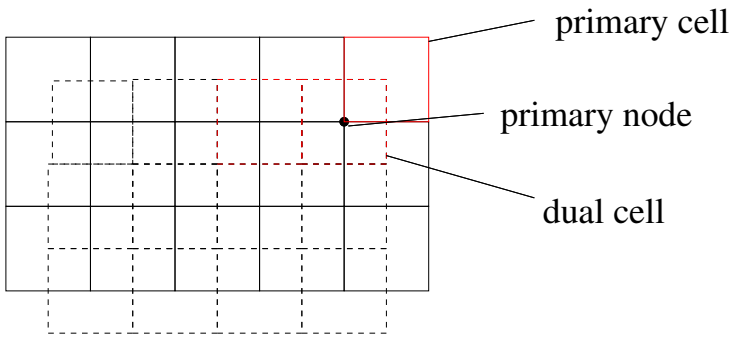

FIGURE 2: Sketch of a primary grid and associated dual grid (dashed lines).

The domain Ω is splitted into the control volumes of the dual grid. We will use the following notations:

- Let $\mathcal{K} = (K_{ij})_{i,j=1}^{N}$ be the set of all control volumes of the dual grid.
- Let $(P_i)_{i=1}^{N}$ be the set of all nodes of the primary grid. P_i and P_j denote neighboring nodes. P_iP_j denotes the edge connecting two neighboring nodes.

In the finite volume method, which is applied in the DLR-THETA-code, velocity and pressure are located at the same nodes. Such grids are called *collocated* grids.

2.4 Collocated finite volume scheme on Cartesian grids

Finite volume methods are applied to solve partial differential equations like the Navier-Stokes equations. The finite volume scheme is a discretization method, i. e., the computational domain is divided into subareas (see section 2.3). At the midpoints of these areas, the unknowns, velocity and pressure are computed.

In this chapter we will present a collocated finite volume scheme on collocated Cartesian grids.

2.4.1 Variational formulation and finite volume method

The finite volume method is derived as follows:

- Splitting Ω into control volumes K_{ij} with $\overline{\Omega} = \cup_{i,j=1}^{n} \overline{K}_{ij}$.
- The test functions of the variational formulation are chosen to be

$$\vec{v} = \sum_{i,j} \vec{v}_{ij}, \quad q = \sum_{i,j} q_{ij},$$

  where $\vec{v}_{ij} = \mathbf{1} := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ and $q_{ij} = 1$ in K_{ij} and zero elsewhere.

- From the variational formulation in (2.8) and (2.9) we obtain with $\vec{v}$ and q

$$\begin{aligned} \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{v}_{ij} \cdot \frac{\partial \vec{u}}{\partial t} dV \; + \; \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{v}_{ij} \cdot \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) dV \\ - \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{v}_{ij} \cdot \nu \Delta \vec{u} dV + \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{v}_{ij} \cdot \vec{\nabla} p dV = \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{v}_{ij} \cdot \vec{f} dV, \\ \sum_{i,j=1}^{n} \int_{K_{ij}} q_{ij} \vec{\nabla} \cdot \vec{u} dV \; = \; 0. \end{aligned}$$

- Applying Gauß' theorem and obtaining by setting $\vec{v}_{ij} = \mathbf{1}$ and $q_{ij} = 1$

$$\begin{aligned} \sum_{i,j=1}^{n} \int_{K_{ij}} \frac{\partial \vec{u}}{\partial t} dV \; + \; \sum_{i,j=1}^{n} \int_{\partial K_{ij}} (\vec{u} \otimes \vec{u}) \vec{n}_{ij} dS \\ - \sum_{i,j=1}^{n} \int_{\partial K_{ij}} \nu \vec{\nabla} \vec{u} \cdot \vec{n}_{ij} dS + \sum_{i,j=1}^{n} \int_{\partial K_{ij}} p \cdot \vec{n}_{ij} dS \; = \sum_{i,j=1}^{n} \int_{K_{ij}} \vec{f} dV \quad (2.14) \\ \sum_{i,j=1}^{n} \int_{\partial K_{ij}} \vec{u} \cdot \vec{n}_{ij} dS \; = \; 0 \end{aligned}$$

  where $\vec{n}_{ij}$ denotes the unit normal vector pointing outward of the boundary ∂K_{ij}.

The result is a coupled system of equations which has to be solved on the dual cells. The finite volume scheme is the governing method of the DLR-THETA-code. In the following sections we will show how the integrals over the dual grid cells are computed.

2.4.2 Formulation of finite volume scheme on collocated Cartesian grids

Our ultimate goal is to solve the 3D Navier-Stokes equations using (2.14) for an incompressible fluid in a bounded domain Ω. For the sake of simplicity we will

consider the 2D case in the further description in an open, rectangular domain $\Omega = (0, L_1) \times (0, L_2)$.

For given volume force $\vec{f} = (f_u, f_v)$, we want to compute the pressure, p and the velocity vector, $\vec{u} = \vec{u}(x, y, t) = (u(x, y, t), v(x, y, t))$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \; - \; \nu\,(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + \frac{\partial p}{\partial x} \quad &= \; f_x \quad \text{in} \quad \Omega \times [0,T], \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \; - \; \nu\,(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) + \frac{\partial p}{\partial y} \quad &= \; f_y \quad \text{in} \quad \Omega \times [0,T], \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \quad &= \; 0 \quad \text{on} \quad \partial\Omega \times [0,T]. \end{aligned} \tag{2.15}$$

The treatment of the boundary of the domain, Ω, will be ignored in this section. We will refer to this later. For the finite volume scheme a local discretization is necessary. Consequently, the domain Ω is discretized by rectangular control volumes. We divide Ω into $M \cdot N$ volumes where $M\Delta x = L_1$ and $N\Delta y = L_2$, such that one control volume has the size $\Delta x \Delta y$. These control volumes are defined by:

$$K_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \quad \text{for} \quad i = 1, \ldots, M \;\; j = 1, \ldots, N,$$

where

$$\begin{aligned} x_{i+\frac{1}{2}} &= i\Delta x \quad \text{for} \quad i = 0, \ldots, M, \\ y_{j+\frac{1}{2}} &= j\Delta y \quad \text{for} \quad j = 0, \ldots, N. \end{aligned}$$

The edges of the control volumes are defined by:

$$\Gamma_{i\pm\frac{1}{2},j} = \left\{(x, y) \;:\; x = x_{i\pm\frac{1}{2}}, y \in [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]\right\} \quad \text{for} \quad i = 0, \ldots, M \quad \text{for vertical edge}$$

and

$$\Gamma_{i,j\pm\frac{1}{2}} = \left\{(x, y) \;:\; x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], y = y_{j\pm\frac{1}{2}}\right\} \quad \text{for} \quad j = 0, \ldots, N \quad \text{for horizontal edge.}$$

As we use a collocated method, velocity and pressure are computed at the center of the cells. To compute an average value for the unknowns which holds at each point, we use the midpoint rule for integrals over a rectangular which dictates that

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \vec{u}(x, y, t)dx\; dy \approx \vec{u}(x_i, y_j, t) \cdot \Delta x\; \Delta y = \vec{u}_{i,j} \cdot \Delta x \cdot \Delta y. \tag{2.16}$$

Using (2.16) we obtain an approximated value for the velocity at each node of the grid.

$$\vec{u}_{i,j}(t) \approx \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \vec{u}(x, y, t)dxdy \quad \text{for the velocity and} \tag{2.17}$$

$$p_{i,j}(t) \approx \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} p(x, y, t)dxdy \quad \text{for the pressure.} \tag{2.18}$$

We also define the velocity fluxes over the edges:

$$F_{u_{i+\frac{1}{2},j}}(t) \approx \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}},y,t)dy \qquad i=0,\dots,M \quad \text{for the horizontal fluxes and} \tag{2.19}$$

$$F_{v_{i,j+\frac{1}{2}}}(t) \approx \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(x,y_{j+\frac{1}{2}},t)dx \qquad j=0,\dots,N \quad \text{for the vertical fluxes.} \tag{2.20}$$

The following sketch illustrates the collocated primary grid:

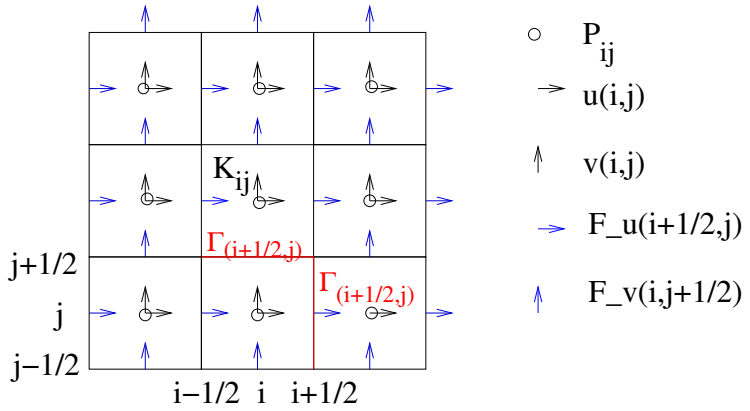

FIGURE 3: Sketch of a collocated grid with primary nodes and velocity field.

2.4.3 Formulation of projection method of van Kan

Now we will focus on the time discretization. We will consider a fixed time intervall, $[0,T]$, and want to compute the unknowns at discrete time steps $t_k = k\Delta t$ where $\Delta t = \frac{T}{N_t}$ and N_t denotes the number of time steps. We define $\vec{u}^k$ as the approximate velocity at time, t_k. From the Navier-Stokes equations ((2.1),(2.2)) we want to determine the velocity, $\vec{u} = (u,v)$ and the pressure, p. As we have three equations, the solutions of $\vec{u}$ and p are determined uniquely; except that p can be shifted by an arbitrary constant.

We will present the time discretization scheme of van Kan ([30]). The problem of (2.15) is that the equations for $\vec{u}$ and p are coupled. The aim is therefore to decouple velocity and pressure. At time step t_{n+1} $\vec{u}^{n+1}$ needs to be computed. The solution $\vec{u}^n$ of the last time step can be also adopted. Therefore, for the time derivative term, $\frac{\partial \vec{u}}{\partial t}$, we use the difference quotient for discretization and obtain,

$$\frac{\partial \vec{u}}{\partial t} \approx \frac{\vec{u}^{n+1}-\vec{u}^n}{\Delta t}.$$

The nonlinear term $(\vec{u}^{n+1}\cdot\vec{\nabla})\vec{u}^{n+1}$ is linearized thus:

$$\left(\vec{u}^{n+1}\cdot\vec{\nabla}\right)\vec{u}^{n+1} \;\approx\; \mathcal{B}\left(\vec{u}^n,\vec{u}^{n-1}\right) := \frac{3}{2}\left(\vec{u}^n\cdot\vec{\nabla}\right)\vec{u}^n - \frac{1}{2}\left(\vec{u}^{n-1}\cdot\vec{\nabla}\right)\vec{u}^{n-1},$$

using the Adams-Bashforth method [3, p.420]. The remaining terms are discretized using the Crank-Nicolson scheme [3, p.531]:

$$\begin{aligned} \nu\Delta\vec{u}^{n+1} &\approx \nu\Delta\frac{\vec{u}^{n+1}+\vec{u}^{n}}{2} \\ \vec{\nabla}p^{n+1} &\approx \vec{\nabla}\frac{p^{n+1}+p^{n}}{2} \\ \vec{f}^{n+1} &\approx \frac{\vec{f}^{n+1}+\vec{f}^{n}}{2} \end{aligned}$$

With these modifications we obtain for the Navier-Stokes equations:

$$\begin{aligned} \frac{\vec{u}^{n+1}-\vec{u}^{n}}{\Delta t}-\nu\Delta(\frac{\vec{u}^{n+1}+\vec{u}^{n}}{2})+\mathcal{B}(\vec{u}^{n},\vec{u}^{n-1})+\vec{\nabla}\frac{p^{n+1}+p^{n}}{2} &= \frac{\vec{f}^{n+1}+\vec{f}^{n}}{2}, \\ \vec{\nabla}\cdot\vec{u}^{n+1} &= 0. \end{aligned}$$

The projection method of van Kan can be divided into the following four steps:

Projection Method of van Kan

Compute the new solution, $\vec{u}^{n+1}$, p^{n+1}, at time step, t_{n+1}, given the solution $\vec{u}^{n}$, p^{n} of the previous time step.

- **step 1**
  Seek $\vec{u}^{n+\frac{1}{2}}$ such that

$$\frac{\vec{u}^{n+\frac{1}{2}}-\vec{u}^{n}}{\Delta t}-\nu\Delta(\frac{\vec{u}^{n+\frac{1}{2}}+\vec{u}^{n}}{2})+\mathcal{B}(\vec{u}^{n},\vec{u}^{n-1})+\vec{\nabla}p^{n}=\frac{\vec{f}^{n+1}+\vec{f}^{n}}{2}. \tag{2.21}$$

- **step 2**
  Seek a pressure correction $\delta p^{n+1}=p^{n+1}-p^{n}$ such that

$$\Delta\delta p^{n+1}=\frac{2}{\Delta t}\vec{\nabla}\cdot\vec{u}^{n+\frac{1}{2}}. \tag{2.22}$$

- **step 3**
  Compute p^{n+1} from δp^{n+1} using

$$p^{n+1}=p^{n}+\delta p^{n+1}. \tag{2.23}$$

- **step 4**
  Compute $\vec{u}^{n+1}$ from

$$\frac{\vec{u}^{n+1}-\vec{u}^{n+\frac{1}{2}}}{\Delta t}=-\frac{1}{2}\vec{\nabla}\delta p^{n+1}. \tag{2.24}$$

N.B. 2.4.1. *From the equations* (2.1) *and* (2.2) *we have three equations and three unknowns* (u, v, p)*. We want to decouple pressure and velocity because the continuity equation does not include the pressure. The system is decoupled by first solving the momentum equation for an intermediate velocity,* $\vec{u}^{n+\frac{1}{2}}$*, using the pressure from the previous time step (step 1).*

- *Generally, the intermediate velocity* $\vec{u}^{n+\frac{1}{2}}$ *does* not *fulfill* $\vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}} = 0$.
- *A pressure correction is computed* such that $\vec{\nabla} \cdot \vec{u}^{n+1} = 0$ *is fulfilled (step 2 and step 3).*
- *In step 4 the velocity based on* $\vec{u}^{n+\frac{1}{2}}$ *from step 1 is projected back onto the space of divergence free functions by updating it using the new pressure.*
- *As a result* $\vec{u}^{n+1}$ *fulfills the continuity equation, i.e.* $\vec{\nabla} \cdot \vec{u}^{n+1} = 0$ *and* $\vec{u}^{n+1}$ *and* p^{n+1} *converge towards a solution of the impulse transport equation for* $\Delta t \to 0$.

We want to prove the results from the projection scheme which are formulated in the following two lemmata.

Lemma 2.4.2. *The velocity* $\vec{u}^{n+1}$ *which is computed with the projection method of van Kan fulfills the continuity equation, i.e.,*

$$\vec{\nabla} \cdot \vec{u}^{n+1} = 0.$$

Proof. From step 4 of the projection scheme we obtain:

$$\begin{aligned} \vec{u}^{n+1} &= -\frac{1}{2} \cdot \vec{\nabla} \delta p^{n+1} \Delta t + \vec{u}^{n+\frac{1}{2}}, \\ \vec{\nabla} \cdot \vec{u}^{n+1} &= -\frac{1}{2} \Delta \delta p^{n+1} \Delta t + \vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}}. \end{aligned} \tag{2.25}$$

From step 2 we obtain for $\vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}}$:

$$\vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}} = \frac{1}{2} \Delta \delta p^{n+1} \Delta t.$$

If we replace $\vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}}$ in (2.25) we obtain:

$$\begin{aligned} \vec{\nabla} \cdot \vec{u}^{n+1} &= -\frac{1}{2} \Delta \delta p^{n+1} \Delta t + \vec{\nabla} \cdot \vec{u}^{n+\frac{1}{2}} \\ &= -\frac{1}{2} \Delta \delta p^{n+1} \Delta t + \frac{1}{2} \Delta \delta p^{n+1} \Delta t \\ &= 0 \end{aligned}$$

So the continuity equation is fulfilled. □

Lemma 2.4.3. *The velocity* $\vec{u}^{n+1}$ *and pressure* p^{n+1} *computed with the projection method of van Kan converge towards a solution of the momentum equation for* $\Delta t \to 0$, *i.e.*

$$\frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} - \nu \Delta \left(\frac{\vec{u}^{n+1} + \vec{u}^n}{2} \right) + \mathcal{B}(\vec{u}^n, \vec{u}^{n-1}) + \vec{\nabla} \frac{p^{n+1} + p^n}{2} \to \frac{\vec{f}^{n+1} + \vec{f}^n}{2} \quad as \quad \Delta t \to 0.$$

Proof. We transform the momentum transport equation by adding zero and using the equation for p^{n+1} from step 3 and obtain for the left hand side of the equation

$$\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}} + \vec{u}^{n+\frac{1}{2}} - \vec{u}^{n}}{\Delta t} - \nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}} + \vec{u}^{n+\frac{1}{2}} + \vec{u}^{n}}{2}) + \mathcal{B}(\vec{u}^{n}, \vec{u}^{n-1}) + \vec{\nabla}\frac{p^{n+1} + p^{n}}{2}.$$

From step 1 we use that $\vec{u}^{n+\frac{1}{2}}$ fulfills

$$\frac{\vec{u}^{n+\frac{1}{2}} - \vec{u}^{n}}{\Delta t} - \nu\Delta(\frac{\vec{u}^{n+\frac{1}{2}} + \vec{u}^{n}}{2}) + \mathcal{B}(\vec{u}^{n}, \vec{u}^{n-1}) + \vec{\nabla}p^{n} = \frac{\vec{f}^{n+1} + \vec{f}^{n}}{2}.$$

Consequently, we have to show that,

$$\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{\Delta t} - \nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{2}) + \frac{1}{2}\vec{\nabla}\delta p^{n+1} \to 0 \qquad \text{as} \qquad \Delta t \to 0.$$

Using step 4 we can substitute the one-sided difference of the time. We obtain:

$$\begin{aligned}
&\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{\Delta t} - \nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{2}) + \frac{1}{2}\vec{\nabla}\delta p^{n+1} \\
=& -\frac{1}{2}\vec{\nabla}\delta p^{n+1} - \nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{2}) + \frac{1}{2}\vec{\nabla}\delta p^{n+1} \\
=& -\nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{2}).
\end{aligned}$$

Substituting the velocity term using step 4 we obtain:

$$-\nu\Delta(\frac{\vec{u}^{n+1} - \vec{u}^{n+\frac{1}{2}}}{2}) = -\nu\Delta(\frac{\vec{\nabla}\delta p^{n+1}\Delta t}{4}).$$

Assuming that the solution $(\vec{u}, p)$ is sufficiently smooth, this converges towards 0 as $\Delta t \to 0$. So the momentum equation is approximately fulfilled if Δt is sufficiently small. □

2.4.4 Finite volume discretization on collocated grids

For each time step we have to solve the equations of the projection scheme for each control volume, K_{ij}, for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. The average value on a control volume is calculated by integration. In this section we want to show how the arising integrals are approximated. In the first step of the projection scheme we have to solve equation (2.21),

$$\frac{\vec{u}^{n+\frac{1}{2}} - \vec{u}^{n}}{\Delta t} - \nu\Delta(\frac{\vec{u}^{n+\frac{1}{2}} + \vec{u}^{n}}{2}) + \mathcal{B}(\vec{u}^{n}, \vec{u}^{n-1}) + \vec{\nabla}p^{n} = \frac{\vec{f}^{n+1} + \vec{f}^{n}}{2}.$$

Discretization of the terms arising in the projection scheme

- We approximate the time derivative term by estimating the integral with the cell-averaged value using the midpoint rule

$$\int_{K_{ij}} \frac{\vec{u}^{n+\frac{1}{2}} - \vec{u}^{n}}{\Delta t} dxdy \approx \Delta x\Delta y \frac{\vec{u}_{i,j}^{n+\frac{1}{2}} - \vec{u}_{i,j}^{n}}{\Delta t}. \tag{2.26}$$

.

- For the Laplace term in (2.21) we apply the Gauß-formula,

$$\int_{K_{ij}} \Delta \vec{u} dx dy = \int_{\partial K_{ij}} \frac{\partial \vec{u}}{\partial n} d\Gamma. \tag{2.27}$$

The integral over the area of K_{ij} is transformed into an integral over the boundary of K_{ij}. Therefore, we have to integrate over the edges of K_{ij} as the following sketch illustrates:

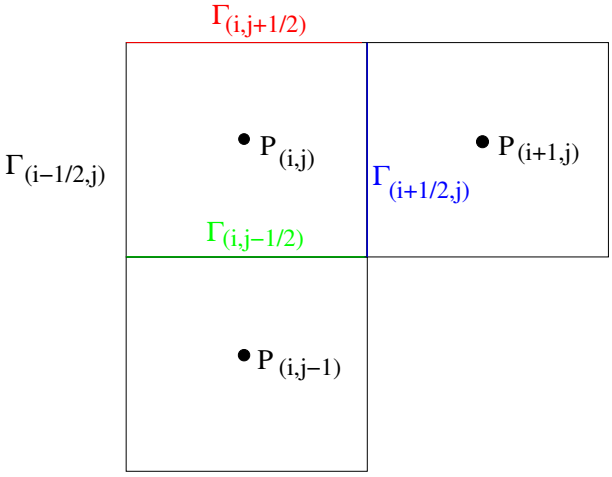

FIGURE 4: Dual cell and its boundary Γ.

We approximate the integral by taking the values of $\vec{u}$ at the midpoints of the edges

$$\begin{aligned}\int_{\partial K_{ij}} \frac{\partial \vec{u}}{\partial n} d\Gamma \;=\; & \int_{\Gamma_{i-\frac{1}{2},j}} (\frac{\partial \vec{u}}{\partial n})_{i-\frac{1}{2},j} d\Gamma + \int_{\Gamma_{i+\frac{1}{2},j}} (\frac{\partial \vec{u}}{\partial n})_{i+\frac{1}{2},j} d\Gamma \\ & + \int_{\Gamma_{i,j-\frac{1}{2}}} (\frac{\partial \vec{u}}{\partial n})_{i,j-\frac{1}{2}} d\Gamma + \int_{\Gamma_{i,j+\frac{1}{2}}} (\frac{\partial \vec{u}}{\partial n})_{i,j+\frac{1}{2}} d\Gamma,\end{aligned}$$

using the following notation

$$\int_{\Gamma} \frac{\partial \vec{u}}{\partial n} d\Gamma := \int_{\Gamma} \vec{\nabla} \vec{u} \cdot \vec{n} d\Gamma = \int_{\Gamma} \sum_{i=1}^{n} \frac{\partial \vec{u}(x)}{\partial x_i} \cdot n_i(x) d\Gamma. \tag{2.28}$$

Approximation of the integrals over the edges is obtained by using the midpoint rule by multiplication with the length of the edge (Δx or Δy) and taking the normal direction in either x- or y-direction according to the definition of (2.28). The partial derivatives are obtained with the first order central difference quotient

$$\begin{aligned}\int_{\partial K_{ij}} \frac{\partial \vec{u}}{\partial n} d\Gamma \quad & \approx \Delta y (\tfrac{\partial \vec{u}}{\partial x})_{i+\frac{1}{2},j} - \Delta y (\tfrac{\partial \vec{u}}{\partial x})_{i-\frac{1}{2},j} \\ & \quad + \Delta x (\tfrac{\partial \vec{u}}{\partial y})_{i,j+\frac{1}{2}} - \Delta x (\tfrac{\partial \vec{u}}{\partial y})_{i,j-\frac{1}{2}} \\ & \approx \Delta y \tfrac{\vec{u}_{i+1,j} - \vec{u}_{i,j}}{\Delta x} + \Delta y \tfrac{\vec{u}_{i-1,j} - \vec{u}_{i,j}}{\Delta x} \\ & \quad + \Delta x \tfrac{\vec{u}_{i,j+1} - \vec{u}_{i,j}}{\Delta y} + \Delta x \tfrac{\vec{u}_{i,j-1} - \vec{u}_{i,j}}{\Delta y}.\end{aligned} \tag{2.29}$$

- Similarly, the Laplacian for δp is written as

$$\begin{aligned}\int_{K_{ij}} \Delta\delta p \; dxdy &= \int_{\partial K_{ij}} \frac{\partial \delta p}{\partial n} d\Gamma \\ &\approx \Delta y \frac{\delta p_{i+1,j} - \delta p_{i,j}}{\Delta x} + \Delta y \frac{\delta p_{i-1,j} - \delta p_{i,j}}{\Delta x} \\ &\quad + \Delta x \frac{\delta p_{i,j+1} - \delta p_{i,j}}{\Delta y} + \Delta x \frac{\delta p_{i,j-1} - \delta p_{i,j}}{\Delta y}.\end{aligned} \tag{2.30}$$

- For the integral of the pressure gradient we first use the Gauß' theorem and obtain a boundary integral which is derived by summing up the integrals over its edges $\Gamma_{i\pm\frac{1}{2},j}$ and $\Gamma_{i,j\pm\frac{1}{2}}$

$$\begin{aligned}\int_{K_{ij}} \vec{\nabla} p dxdy &= \int_{\partial K_{ij}} p \cdot \vec{n} \; d\Gamma \\ &= \int_{\Gamma_{i+\frac{1}{2},j}} p\vec{n} d\Gamma_{i+\frac{1}{2},j} + \int_{\Gamma_{i-\frac{1}{2},j}} p\vec{n} d\Gamma_{i-\frac{1}{2},j} \\ &\quad + \int_{\Gamma_{i,j+\frac{1}{2}}} p\vec{n} d\Gamma_{i,j+\frac{1}{2}} + \int_{\Gamma_{i,j-\frac{1}{2}}} p\vec{n} d\Gamma_{i,j-\frac{1}{2}}.\end{aligned} \tag{2.31}$$

These integrals are obtained by averaging the value at the midpoints of the edge, i.e. we use a central difference scheme.

$$\begin{aligned}\int_{\Gamma_{i,j-\frac{1}{2}}} p\vec{n} d\Gamma_{i,j-\frac{1}{2}} &\approx \Delta x \begin{pmatrix} 0 \\ -\frac{1}{2}(p_{i,j} + p_{i,j-1}) \end{pmatrix}, \\ \int_{\Gamma_{i,j+\frac{1}{2}}} p\vec{n} d\Gamma_{i,j+\frac{1}{2}} &\approx \Delta x \begin{pmatrix} 0 \\ \frac{1}{2}(p_{i,j} + p_{i,j+1}) \end{pmatrix}, \\ \int_{\Gamma_{i+\frac{1}{2},j}} p\vec{n} d\Gamma_{i+\frac{1}{2},j} &\approx \Delta y \begin{pmatrix} \frac{1}{2}(p_{i,j} + p_{i+1,j}) \\ 0 \end{pmatrix}, \\ \int_{\Gamma_{i-\frac{1}{2},j}} p\vec{n} d\Gamma_{i-\frac{1}{2},j} &\approx \Delta y \begin{pmatrix} -\frac{1}{2}(p_{i,j} + p_{i-1,j}) \\ 0 \end{pmatrix}.\end{aligned}$$

We obtain by summing up the integrals

$$\int_{K_{ij}} \vec{\nabla} p \; dxdy = \begin{pmatrix} \frac{\Delta y}{2}(p_{i+1,j} - p_{i-1,j}) \\ \frac{\Delta x}{2}(p_{i,j+1} - p_{i,j-1}) \end{pmatrix}. \tag{2.32}$$

- The linearized term of Adams-Bashforth extrapolation can be approximated by applying the Gauß' theorem and using linear interpolation for the velocity at the midpoints of the edges. The terms of horizontal and vertical fluxes are

taken into account, substituting part of the integrals over the edges.

$$\begin{aligned}
\int_{K_{ij}} \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) dxdy &= \int_{K_{ij}} \begin{pmatrix} \partial_x(u^2) + \partial_y(vu) \\ \partial_x(uv) + \partial_y(v^2) \end{pmatrix} dxdy \\
&= \int_{K_{ij}} \vec{\nabla} \cdot \begin{pmatrix} u^2 & vu \\ uv & v^2 \end{pmatrix} dxdy \\
&= \int_{\partial K_{ij}} \begin{pmatrix} u^2 & vu \\ uv & v^2 \end{pmatrix} \vec{n} d\Gamma \\
&= \int_{\partial K_{ij}} \begin{pmatrix} (u^2)n_x + (vu)n_y \\ (uv)n_x + (v^2)n_y \end{pmatrix} d\Gamma \\
&\approx \begin{pmatrix} \Delta y F_{u_{i+\frac{1}{2},j}} \frac{u_{i+1,j}+u_{i,j}}{2} - \Delta y F_{u_{i-\frac{1}{2},j}} \frac{u_{i,j}+u_{i-1,j}}{2} \\ \Delta y F_{u_{i+\frac{1}{2},j}} \frac{v_{i+1,j}+v_{i,j}}{2} - \Delta y F_{u_{i-\frac{1}{2},j}} \frac{v_{i,j}+v_{i-1,j}}{2} \end{pmatrix} \\
&\quad + \begin{pmatrix} \Delta x F_{v_{i,j+\frac{1}{2}}} \frac{u_{i,j+1}+u_{i,j}}{2} - \Delta x F_{v_{i,j-\frac{1}{2}}} \frac{u_{i,j}+u_{i,j-1}}{2} \\ \Delta x F_{v_{i,j+\frac{1}{2}}} \frac{v_{i,j+1}+v_{i,j}}{2} - \Delta x F_{v_{i,j-\frac{1}{2}}} \frac{v_{i,j}+v_{i,j-1}}{2} \end{pmatrix}.
\end{aligned} \tag{2.33}$$

- For the divergence term we use Gauß' theorem and the terms for the fluxes at the edges

$$\begin{aligned}
\int_{K_{ij}} \vec{\nabla} \cdot \vec{u} dxdy &= \int_{\partial K_{ij}} \vec{u} \cdot \vec{n} d\Gamma \\
&\approx \Delta y (F_{u_{i+\frac{1}{2},j}} - F_{u_{i-\frac{1}{2},j}}) + \Delta x (F_{v_{i,j+\frac{1}{2}}} - F_{v_{i,j-\frac{1}{2}}}).
\end{aligned} \tag{2.34}$$

Step 1: Discretization of the equation for intermediate velocity

To compute the intermediate velocity $\vec{u}^{n+\frac{1}{2}}$ in the projection scheme we obtain the following equation:

$$\begin{aligned}
&\int_{K_{ij}} \frac{\vec{u}^{n+\frac{1}{2}} - \vec{u}^n}{\Delta t} dxdy - \int_{K_{ij}} \nu \Delta (\frac{\vec{u}^{n+\frac{1}{2}} + \vec{u}^n}{2}) dxdy \\
&+ \int_{K_{ij}} \mathcal{B}(\vec{u}^n, \vec{u}^{n-1}) dxdy + \int_{K_{ij}} \vec{\nabla} p^n dxdy \\
&= \int_{K_{ij}} \frac{\vec{f}^{n+1} + \vec{f}^n}{2} dxdy.
\end{aligned} \tag{2.35}$$

- By approximation of the time derivative term using (2.26) we obtain:

$$\int_{K_{ij}} \frac{\vec{u}_{i,j}^{n+\frac{1}{2}} - \vec{u}_{i,j}^n}{\Delta t} dxdy \approx \Delta x \Delta y \frac{\vec{u}_{i,j}^{n+\frac{1}{2}} - \vec{u}_{i,j}^n}{\Delta t}. \tag{2.36}$$

- Using (2.27) and (2.29) we obtain for the Laplace term:

$$\begin{aligned}\int_{K_{ij}} \nu\Delta(\frac{\vec{u}^{n+\frac{1}{2}}+\vec{u}^{n}}{2})dxdy \approx \frac{\nu}{2}[&\Delta y\frac{\vec{u}^{n+\frac{1}{2}}_{i+1,j}-\vec{u}^{n+\frac{1}{2}}_{i,j}}{\Delta x}+\Delta y\frac{\vec{u}^{n+\frac{1}{2}}_{i-1}-\vec{u}^{n+\frac{1}{2}}_{i,j}}{\Delta x}\\ &+\Delta x\frac{\vec{u}^{n+\frac{1}{2}}_{i,j+1}-\vec{u}^{n+\frac{1}{2}}_{i,j}}{\Delta y}+\Delta x\frac{\vec{u}^{n+\frac{1}{2}}_{i,j-1}-\vec{u}^{n+\frac{1}{2}}_{i,j}}{\Delta y}\\ &+\Delta y\frac{\vec{u}^{n}_{i+1,j}-\vec{u}^{n}_{i,j}}{\Delta x}+\Delta y\frac{\vec{u}^{n}_{i-1,j}-\vec{u}^{n}_{i,j}}{\Delta x}\\ &+\Delta x\frac{\vec{u}^{n}_{i,j+1}-\vec{u}^{n}_{i,j}}{\Delta y}+\Delta x\frac{\vec{u}^{n}_{i,j-1}-\vec{u}^{n}_{i,j}}{\Delta y}].\end{aligned}$$

.

- In order to approximate the nonlinear term $\mathcal{B}(\vec{u}^n,\vec{u}^{n-1})$ we use (2.33) and obtain

$$\begin{aligned}\int_{K_{ij}} \frac{3}{2}(\vec{u}^n\vec{\nabla})\vec{u}^n dxdy \approx &\frac{3}{2}[\Delta y F^n_{u_{i+\frac{1}{2},j}}\frac{\vec{u}^n_{i+1,j}+\vec{u}^n_{i,j}}{2}-\Delta y F^n_{u_{i-\frac{1}{2},j}}\frac{\vec{u}^n_{i,j}+\vec{u}^n_{i-1,j}}{2}\\ &+\Delta x F^n_{v_{i,j+\frac{1}{2}}}\frac{\vec{u}^n_{i,j+1}+\vec{u}^n_{i,j}}{2}-\Delta x F^n_{v_{i,j-\frac{1}{2}}}\frac{\vec{u}^n_{i,j}+\vec{u}^n_{i,j-1}}{2}],\\ \int_{K_{ij}} \frac{1}{2}(\vec{u}^{n-1}\vec{\nabla})\vec{u}^{n-1} dxdy \approx &\frac{1}{2}[\Delta y F^{n-1}_{u_{i+\frac{1}{2},j}}\frac{\vec{u}^{n-1}_{i+1,j}+\vec{u}^{n-1}_{i,j}}{2}-\Delta y F^{n-1}_{u_{i-\frac{1}{2},j}}\frac{\vec{u}^{n-1}_{i,j}+\vec{u}^{n-1}_{i-1,j}}{2}\\ &+\Delta x F^{n-1}_{v_{i,j+\frac{1}{2}}}\frac{\vec{u}^{n-1}_{i,j+1}+\vec{u}^{n-1}_{i,j}}{2}-\Delta x F^{n-1}_{v_{i,j-\frac{1}{2}}}\frac{\vec{u}^{n-1}_{i,j}+\vec{u}^{n-1}_{i,j-1}}{2}].\end{aligned}$$

.

- The pressure gradient, $\vec{\nabla}p$, is approximated using (2.31):

$$\int_{K_{ij}} \vec{\nabla}p dxdy \approx \begin{pmatrix}\frac{\Delta y}{2}(p_{i+1,j}-p_{i-1,j})\\ \frac{\Delta x}{2}(p_{i,j+1}-p_{i,j-1})\end{pmatrix}.$$

.

- The integral of the body force is approximated using the mean value over the cell face area:

$$\int_{K_{ij}} \frac{\vec{f}^{n+1}+\vec{f}^{n}}{2}dxdy \approx \Delta x\Delta y\frac{\vec{f}^{n+1}_{i,j}+\vec{f}^{n}_{i,j}}{2}.$$

Step 2: Discretization of the equation for the pressure correction

We solve the equation for the pressure correction

$$\Delta\delta p^{n+1} = \frac{2}{\Delta t}\vec{\nabla}\cdot\vec{u}^{n+\frac{1}{2}},$$

by using (2.30) and (2.34) and obtain

$$\begin{aligned}\Delta y\frac{\delta p^{n+1}_{i+1,j}-\delta p^{n+1}_{i,j}}{\Delta x}+\Delta y\frac{\delta p^{n+1}_{i-1,j}-\delta p^{n+1}_{i,j}}{\Delta x}+\Delta x\frac{\delta p^{n+1}_{i,j+1}-\delta p^{n+1}_{i,j}}{\Delta y}+\Delta x\frac{\delta p^{n+1}_{i,j-1}-\delta p^{n+1}_{i,j}}{\Delta y}\\ =\frac{2}{\Delta t}[\Delta y(F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}}-F^{n+\frac{1}{2}}_{u_{i-\frac{1}{2},j}})+\Delta x(F^{n+\frac{1}{2}}_{v_{i,j+\frac{1}{2}}}-F^{n+\frac{1}{2}}_{v_{i,j-\frac{1}{2}}})].\end{aligned} \tag{2.37}$$

Step 3: Computation of divergence free velocity

Compute the new velocity, $\vec{u}^{n+1}$, and the new fluxes, F_u^{n+1} and F_v^{n+1}, using (2.24) and the approximation for the pressure gradient (2.31)

$$\begin{aligned}
\Delta x \Delta y \frac{\vec{u}_{i,j}^{n+1} - \vec{u}_{i,j}^{n+\frac{1}{2}}}{\Delta t} &= -\frac{1}{2}\begin{pmatrix} \frac{\Delta y}{2}(\delta p_{i+1,j}^{n+1} - \delta p_{i-1,j}^{n+1}) \\ \frac{\Delta x}{2}(\delta p_{i,j+1}^{n+1} - \delta p_{i,j-1}^{n+1}) \end{pmatrix}, \\
\frac{F_{u_{i+\frac{1}{2},j}}^{n+1} - F_{u_{i+\frac{1}{2},j}}^{n+\frac{1}{2}}}{\Delta t} &= -\frac{1}{2}\frac{\delta p_{i+1,j}^{n+1} - \delta p_{i,j}^{n+1}}{\Delta x}, \\
\frac{F_{v_{i,j+\frac{1}{2}}}^{n+1} - F_{v_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}}}{\Delta t} &= -\frac{1}{2}\frac{\delta p_{i,j+1}^{n+1} - \delta p_{i,j}^{n+1}}{\Delta y}.
\end{aligned}$$

2.4.5 Interpretation of horizontal and vertical fluxes

For the following considerations the intermediate fluxes play an important role. We will only consider the horizontal fluxes as the interpretation of the vertical fluxes follows analogously. The horizontal flux is defined as:

$$F_{u_{i+\frac{1}{2},j}}(t) \approx \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}}, y, t)dy. \tag{2.38}$$

The integral is computed over the vertical edge $\Gamma_{i+\frac{1}{2},j}$. A flux over an area is defined as the integral of the velocity field over this area. Consequently, the horizontal flux over the vertical edge is the integral over $\Gamma_{i+\frac{1}{2},j}$. In the following sketch the horizontal flux over the edge $\Gamma_{i+\frac{1}{2},j}$ is illustrated.

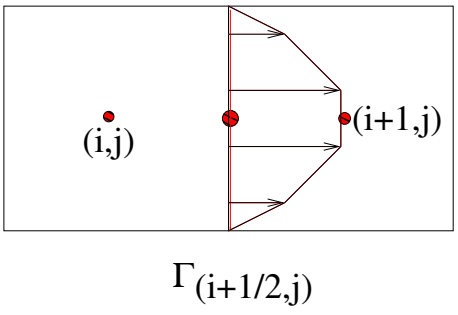

FIGURE 5: The integral of the velocity field over the vertical edge $\Gamma_{i+\frac{1}{2},j}$.

The value of the velocity at $x_{i+\frac{1}{2}}$ is unknown as it is the midpoint of the edge $\Gamma_{i+\frac{1}{2},j}$ and does not belong to the set of the primary nodes. The horizontal fluxes over the edges $\Gamma_{i,j}$ and $\Gamma_{i+1,j}$ can easily be computed as the value for the velocity is constant within each control volume K_{ij} and the values $\vec{u}_{i,j}$ and $\vec{u}_{i+1,j}$ at the primary nodes are known. Consequently, the velocity at $\Gamma_{i,j}$ is equal to $\vec{u}_{i,j}$ and analogously at $\Gamma_{i+1,j}$ it is equal to $\vec{u}_{i+1,j}$. The following sketch provides some illustration:

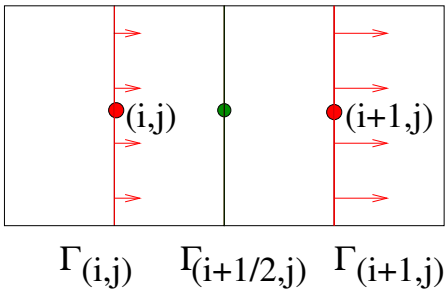

FIGURE 6: Illustration of the approximated velocity field over the vertical edges.

The vertical flux over the edge $\Gamma_{i+\frac{1}{2},j}$ can now be derived using linear interpolation,

$$\begin{aligned} F_{u_{i+\frac{1}{2},j}}(t) \approx & \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}}, y, t) dy \\ = & \frac{1}{2\Delta y} \left(\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_i, y, t) dy + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+1}, y, t) dy \right). \end{aligned}$$

The following sketch illustrates the computation of the integral at $\Gamma_{i+\frac{1}{2},j}$:

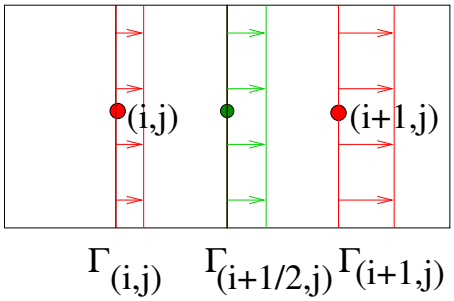

FIGURE 7: The horizontal fluxes over the edges. The integral at $\Gamma_{i+\frac{1}{2},j}$ is the average value of the integrals at (i, j) and $(i + 1, j)$.

The integrals over the vertical edges $\Gamma_{i,j}$ and $\Gamma_{i+1,j}$ can be computed using the midpoint rule:

$$\begin{aligned} \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} u(x_i, y, t) dy \approx & \ \Delta y \ u_{i,j}, \\ \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} u(x_{i+1}, y, t) dy \approx & \ \Delta y \ u_{i+1,j}. \end{aligned}$$

Consequently, the vertical flux is an interpolation for the velocity at the midpoint of the edge $\Gamma_{i+\frac{1}{2},j}$ between two primary nodes (i,j) and $(i+1,j)$ as

$$\begin{aligned} F_{u_{i+\frac{1}{2},j}}(t) &\approx \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}},y,t)dy \\ &\approx \frac{1}{2\Delta y}\left(\int_{j-\frac{1}{2}}^{j+\frac{1}{2}} u(x_i,y,t)dy + \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} u(x_{i+1},y,t)dy\right) \\ &\approx \frac{u_{i,j}+u_{i+1,j}}{2} \approx u_{i+\frac{1}{2},j}. \end{aligned}$$

2.4.6 Computation of intermediate fluxes

In this section we give attention to the computation of the intermediate fluxes. The method of interpolation is essential because it determines the coupling between velocity and pressure. The simplest way of computation is the linear interpolation for

$$\text{horizontal fluxes} \quad F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}} \approx \frac{u^{n+\frac{1}{2}}_{i+1,j}+u^{n+\frac{1}{2}}_{i,j}}{2}, \tag{2.39}$$

$$\text{vertical fluxes} \quad F^{n+\frac{1}{2}}_{v_{i,j+\frac{1}{2}}} \approx \frac{v^{n+\frac{1}{2}}_{i,j+1}+v^{n+\frac{1}{2}}_{i,j}}{2}. \tag{2.40}$$

For pragmatic reasons we will only consider the horizontal fluxes. For the linear interpolation the intermediate velocity $u^{n+\frac{1}{2}}$ is essential. From (2.35) using the discretization methods of (2.26)-(2.31) we obtain:

$$\begin{aligned} &\Delta x\Delta y\frac{u^{n+\frac{1}{2}}_{i,j}-u^n_{i,j}}{\Delta t} - \frac{\nu}{2}[\Delta y\frac{u^{n+\frac{1}{2}}_{i+1,j}-u^{n+\frac{1}{2}}_{i,j}}{\Delta x} + \Delta y\frac{u^{n+\frac{1}{2}}_{i-1,j}-u^{n+\frac{1}{2}}_{i,j}}{\Delta x} + \Delta x\frac{u^{n+\frac{1}{2}}_{i,j+1}-u^{n+\frac{1}{2}}_{i,j}}{\Delta y} \\ &+ \Delta x\frac{u^{n+\frac{1}{2}}_{i,j-1}-u^{n+\frac{1}{2}}_{i,j}}{\Delta y} + \Delta y\frac{u^n_{i+1,j}-u^n_{i,j}}{\Delta x} + \Delta y\frac{u^n_{i-1,j}-u^n_{i,j}}{\Delta x} + \Delta x\frac{u^n_{i,j+1}-u^n_{i,j}}{\Delta y} \\ &+ \Delta x\frac{u^n_{i,j-1}-u^n_{i,j}}{\Delta y}] + \frac{3}{2}[\Delta y F^n_{u_{i+\frac{1}{2},j}}\frac{u^n_{i+1,j}+u^n_{i,j}}{2} - \Delta y F^n_{u_{i-\frac{1}{2},j}}\frac{u^n_{i,j}+u^n_{i-1,j}}{2} \\ &+ \Delta x F^n_{v_{i,j+\frac{1}{2}}}\frac{u^n_{i,j+1}+u^n_{i,j}}{2} - \Delta x F^n_{v_{i,j-\frac{1}{2}}}\frac{u^n_{i,j}+u^n_{i,j-1}}{2}] \\ &- \frac{1}{2}[\Delta y F^{n-1}_{u_{i+\frac{1}{2},j}}\frac{u^{n-1}_{i+1,j}+u^{n-1}_{i,j}}{2} - \Delta y F^{n-1}_{u_{i-\frac{1}{2},j}}\frac{u^{n-1}_{i,j}+u^{n-1}_{i-1,j}}{2} \\ &+ \Delta x F^{n-1}_{v_{i,j+\frac{1}{2}}}\frac{u^{n-1}_{i,j+1}+u^{n-1}_{i,j}}{2} - \Delta x F^{n-1}_{v_{i,j-\frac{1}{2}}}\frac{u^{n-1}_{i,j}+u^{n-1}_{i,j-1}}{2}] + \frac{\Delta y}{2}\left(p^n_{i+1,j}-p^n_{i-1,j}\right) \\ &= \Delta x\Delta y\frac{f^{n+1}_{i,j}+f^n_{i,j}}{2}. \end{aligned}$$

Now we will consider all terms which contain $u^{n+\frac{1}{2}}_{i,j}$

$$\Delta y\frac{u^{n+\frac{1}{2}}_{i+1,j}-u^{n+\frac{1}{2}}_{i,j}}{\Delta x}, \qquad \Delta y\frac{u^{n+\frac{1}{2}}_{i-1,j}-u^{n+\frac{1}{2}}_{i,j}}{\Delta x}, \qquad \Delta x\frac{u^{n+\frac{1}{2}}_{i,j+1}-u^{n+\frac{1}{2}}_{i,j}}{\Delta y}, \qquad \Delta x\frac{u^{n+\frac{1}{2}}_{i,j-1}-u^{n+\frac{1}{2}}_{i,j}}{\Delta y}.$$

We will separate the terms containing $u_{i,j}^{n+\frac{1}{2}}$ in order to isolate the intermediate velocity,

$$\begin{aligned}
&\Delta x\Delta y\frac{u_{i,j}^{n+\frac{1}{2}}}{\Delta t}-\frac{\nu}{2}[-\Delta y\frac{u_{i,j}^{n+\frac{1}{2}}}{\Delta x}-\Delta y\frac{u_{i,j}^{n+\frac{1}{2}}}{\Delta x}-\Delta x\frac{u_{i,j}^{n+\frac{1}{2}}}{\Delta y}-\Delta x\frac{u_{i,j}^{n+\frac{1}{2}}}{\Delta y}].\\
&-\Delta x\Delta y\frac{u_{i,j}^{n}}{\Delta t}-\frac{\nu}{2}[\Delta y\frac{u_{i+1,j}^{n+\frac{1}{2}}}{\Delta x}+\Delta y\frac{u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x}+\Delta x\frac{u_{i,j+1}^{n+\frac{1}{2}}}{\Delta y}+\Delta x\frac{u_{i,j-1}^{n+\frac{1}{2}}}{\Delta y}\\
&+\Delta y\frac{u_{i+1,j}^{n}-u_{i,j}^{n}}{\Delta x}+\Delta y\frac{u_{i-1,j}^{n}-u_{i,j}^{n}}{\Delta x}+\Delta x\frac{u_{i,j+1}^{n}-u_{i,j}^{n}}{\Delta y}+\Delta x\frac{u_{i,j-1}^{n}-u_{i,j}^{n}}{\Delta y}]\\
&+\frac{3}{2}[\Delta yF_{u_{i+\frac{1}{2},j}}^{n}\frac{u_{i+1,j}^{n}+u_{i,j}^{n}}{2}-\Delta yF_{u_{i-\frac{1}{2},j}}^{n}\frac{u_{i,j}^{n}+u_{i-1,j}^{n}}{2}\\
&+\Delta xF_{v_{i,j+\frac{1}{2}}}^{n}\frac{u_{i,j+1}^{n}+u_{i,j}^{n}}{2}-\Delta xF_{v_{i,j-\frac{1}{2}}}^{n}\frac{u_{i,j}^{n}+u_{i,j-1}^{n}}{2}]\\
&-\frac{1}{2}[\Delta yF_{u_{i+\frac{1}{2},j}}^{n-1}\frac{u_{i+1,j}^{n-1}+u_{i,j}^{n-1}}{2}-\Delta yF_{u_{i-\frac{1}{2},j}}^{n-1}\frac{u_{i,j}^{n-1}+u_{i-1,j}^{n-1}}{2}\\
&+\Delta xF_{v_{i,j+\frac{1}{2}}}^{n-1}\frac{u_{i,j+1}^{n-1}+u_{i,j}^{n-1}}{2}-\Delta xF_{v_{i,j-\frac{1}{2}}}^{n-1}\frac{u_{i,j}^{n-1}+u_{i,j-1}^{n-1}}{2}]\\
&+\frac{\Delta y}{2}\left(p_{i+1,j}^{n}-p_{i-1,j}^{n}\right)\\
&=\Delta x\Delta y\frac{f_{i,j}^{n+1}+f_{i,j}^{n}}{2}.
\end{aligned}$$

Factoring out $u_{i,j}^{n+\frac{1}{2}}$ we obtain:

$$\begin{aligned}
&u_{i,j}^{n+\frac{1}{2}}[\frac{\Delta x\Delta y}{\Delta t}+\nu\frac{\Delta y}{\Delta x}+\nu\frac{\Delta x}{\Delta y}]\\
&-\Delta x\Delta y\frac{u_{i,j}^{n}}{\Delta t}-\frac{\nu}{2}[\Delta y\frac{u_{i+1,j}^{n+\frac{1}{2}}}{\Delta x}+\Delta y\frac{u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x}+\Delta x\frac{u_{i,j+1}^{n+\frac{1}{2}}}{\Delta y}+\Delta x\frac{u_{i,j-1}^{n+\frac{1}{2}}}{\Delta y}\\
&+\Delta y\frac{u_{i+1,j}^{n}-u_{i,j}^{n}}{\Delta x}+\Delta y\frac{u_{i-1,j}^{n}-u_{i,j}^{n}}{\Delta x}+\Delta x\frac{u_{i,j+1}^{n}-u_{i,j}^{n}}{\Delta y}+\Delta x\frac{u_{i,j-1}^{n}-u_{i,j}^{n}}{\Delta y}]\\
&+\frac{3}{2}[\Delta yF_{u_{i+\frac{1}{2},j}}^{n}\frac{u_{i+1,j}^{n}+u_{i,j}^{n}}{2}-\Delta yF_{u_{i-\frac{1}{2},j}}^{n}\frac{u_{i,j}^{n}+u_{i-1,j}^{n}}{2}\\
&+\Delta xF_{v_{i,j+\frac{1}{2}}}^{n}\frac{u_{i,j+1}^{n}+u_{i,j}^{n}}{2}-\Delta xF_{v_{i,j-\frac{1}{2}}}^{n}\frac{u_{i,j}^{n}+u_{i,j-1}^{n}}{2}]\\
&-\frac{1}{2}[\Delta yF_{u_{i+\frac{1}{2},j}}^{n-1}\frac{u_{i+1,j}^{n-1}+u_{i,j}^{n-1}}{2}-\Delta yF_{u_{i-\frac{1}{2},j}}^{n-1}\frac{u_{i,j}^{n-1}+u_{i-1,j}^{n-1}}{2}\\
&+\Delta xF_{v_{i,j+\frac{1}{2}}}^{n-1}\frac{u_{i,j+1}^{n-1}+u_{i,j}^{n-1}}{2}-\Delta xF_{v_{i,j-\frac{1}{2}}}^{n-1}\frac{u_{i,j}^{n-1}+u_{i,j-1}^{n-1}}{2}]\\
&+\frac{\Delta y}{2}\left(p_{i+1,j}^{n}-p_{i-1,j}^{n}\right)\\
&=\Delta x\Delta y\frac{f_{i,j}^{n+1}+f_{i,j}^{n}}{2}.
\end{aligned}$$

To isolate the intermediate velocity we shift the terms not containing it to the right and divide by:

$$a:=[\frac{\Delta x\Delta y}{\Delta t}+\nu\frac{\Delta y}{\Delta x}+\nu\frac{\Delta x}{\Delta y}].$$

We obtain:

$$
\begin{aligned}
u_{i,j}^{n+\frac{1}{2}} =& \frac{1}{a}\{\Delta x\Delta y\frac{u_{i,j}^n}{\Delta t} + \frac{\nu}{2}[\Delta y\frac{u_{i+1,j}^{n+\frac{1}{2}}}{\Delta x} + \Delta y\frac{u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x} + \Delta x\frac{u_{i,j+1}^{n+\frac{1}{2}}}{\Delta y} + \Delta x\frac{u_{i,j-1}^{n+\frac{1}{2}}}{\Delta y} \\
&+ \Delta y\frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x} + \Delta y\frac{u_{i-1,j}^n - u_{i,j}^n}{\Delta x} + \Delta x\frac{u_{i,j+1}^n - u_{i,j}^n}{\Delta y} + \Delta x\frac{u_{i,j-1}^n - u_{i,j}^n}{\Delta y}] \\
&- \frac{3}{2}[\Delta y F_{u_{i+\frac{1}{2},j}}^n\frac{u_{i+1,j}^n + u_{i,j}^n}{2} - \Delta y F_{u_{i-\frac{1}{2},j}}^n\frac{u_{i,j}^n + u_{i-1,j}^n}{2} \\
&+ \Delta x F_{v_{i,j+\frac{1}{2}}}^n\frac{u_{i,j+1}^n + u_{i,j}^n}{2} - \Delta x F_{v_{i,j-\frac{1}{2}}}^n\frac{u_{i,j}^n + u_{i,j-1}^n}{2}] \\
&+ \frac{1}{2}[\Delta y F_{u_{i+\frac{1}{2},j}}^{n-1}\frac{u_{i+1,j}^{n-1} + u_{i,j}^{n-1}}{2} - \Delta y F_{u_{i-\frac{1}{2},j}}^{n-1}\frac{u_{i,j}^{n-1} + u_{i-1,j}^{n-1}}{2} \\
&+ \Delta x F_{v_{i,j+\frac{1}{2}}}^{n-1}\frac{u_{i,j+1}^{n-1} + u_{i,j}^{n-1}}{2} - \Delta x F_{v_{i,j-\frac{1}{2}}}^{n-1}\frac{u_{i,j}^{n-1} + u_{i,j-1}^{n-1}}{2}] + \Delta x\Delta y\frac{f_{i,j}^{n+1} + f_{i,j}^n}{2}\} \\
&- \frac{1}{a}\frac{\Delta y}{2}\left(p_{i+1,j}^n - p_{i-1,j}^n\right).
\end{aligned}
\tag{2.41}
$$

2.4.7 Problem: Checkerboard effect

If we only consider the *last term* in the right-hand side of equation (2.41), which gives the contribution of the pressure to the intermediate velocity $u_{ij}^{n+\frac{1}{2}}$, we obtain:

$$
u_{ij}^{n+\frac{1}{2}} = \mathcal{G}_1 - \frac{1}{a}\frac{\Delta y}{2}\left(p_{i+1,j}^n - p_{i-1,j}^n\right),
$$

with $\mathcal{G}_1$ including all terms depending on the intermediate velocities $u^{n+\frac{1}{2}}$, the velocities u^n and the fluxes F^n from the previous step. Considering the computation of the horizontal flux (2.39) we obtain the following term

$$
F_{ui+\frac{1}{2},j}^{n+\frac{1}{2}} = \frac{u_{i+1,j}^{n+\frac{1}{2}} + u_{i,j}^{n+\frac{1}{2}}}{2} = \mathcal{G}_2 - \frac{1}{a}\frac{\Delta y}{4}\left(p_{i+2,j}^n + p_{i+1,j}^n - p_{i,j}^n - p_{i-1,j}^n\right), \tag{2.42}
$$

with $\mathcal{G}_2$ including all terms which depend on u^n, $u^{n+\frac{1}{2}}$ and F^n. In the second step of the projection scheme (2.37) the Laplacian of the pressure is computed using the difference of the horizontal and vertical intermediate fluxes

$$
\Delta\delta p^{n+1} = \frac{2}{\Delta t}[\Delta y(F_{u_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} - F_{u_{i-\frac{1}{2},j}}^{n+\frac{1}{2}}) + \Delta x(F_{v_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}} - F_{v_{i,j-\frac{1}{2}}}^{n+\frac{1}{2}})]. \tag{2.43}
$$

Computing the difference with the interpolation scheme we used above we obtain for the horizontal and vertical fluxes the following:

$$
\begin{aligned}
F_{u_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} - F_{u_{i-\frac{1}{2},j}}^{n+\frac{1}{2}} &= \frac{u_{i+1,j}^{n+\frac{1}{2}} + u_{i,j}^{n+\frac{1}{2}}}{2} - \frac{u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{2} \\
&= \frac{u_{i+1,j}^{n+\frac{1}{2}} - u_{i-1,j}^{n+\frac{1}{2}}}{2}, \\
F_{v_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}} - F_{v_{i,j-\frac{1}{2}}}^{n+\frac{1}{2}} &= \frac{v_{i,j+1}^{n+\frac{1}{2}} + v_{i,j}^{n+\frac{1}{2}}}{2} - \frac{v_{i,j}^{n+\frac{1}{2}} + v_{i,j-1}^{n+\frac{1}{2}}}{2} \\
&= \frac{v_{i,j+1}^{n+\frac{1}{2}} - v_{i,j-1}^{n+\frac{1}{2}}}{2}.
\end{aligned}
$$

Using the formula (2.42) and (2.43) we obtain for the computation of the Laplacian of the pressure

$$\Delta\delta p^{n+1} = \frac{2}{\Delta t}\{\mathcal{G} + \Delta y\left(p^n_{i+2,j} - 2p^n_{i,j} + p^n_{i-2,j}\right) + \Delta x\left(p^n_{i,j+2} - 2p^n_{i,j} + p^n_{i,j-2}\right)\}, \tag{2.44}$$

with $\mathcal{G}$ including all terms which depend on the intermediate velocities $\vec{u}^{n+\frac{1}{2}}$, the velocities $\vec{u}^n$ and the fluxes F^n from the step before. We come to the conclusion that the computation of $\delta\Delta p^{n+1}_{ij}$ does *not* depend on the values of the pressure at the neighboring nodes.

N.B. 2.4.4. *In the second step of the projection scheme the pressure term $\delta p^{n+1}_{i,j}$ is computed from the divergence of the intermediate velocity, $u^{n+\frac{1}{2}}_{i,j}$. Using the discretization and interpolation methods above, the contribution of the pressure in* (2.44) *shows that the pressure terms, $p^n_{i+1,j}$ and $p^n_{i-1,j}$, are irrelevant for the divergence of the intermediate velocity. The consequence thereof is that the new velocity, $u^{n+1}_{i,j}$, is computed without taking $p^n_{i-1,j}$ and $p^n_{i+1,j}$ into account . Such an interpolation leads to incorrect numerical results when we increase the Reynolds number. The effect of this interpolation is called the* checkerboard effect *[7].*

2.4.8 Approach to avoid the checkerboard effect

The checkerboard effect is remedied using an approach proposed by Rhie and Chow [22]. This approach modifies the interpolation scheme used for the intermediate fluxes. Let us now describe how we will compute the intermediate fluxes according to Rhie and Chow. For the intermediate fluxes we use the following interpolation scheme:

$$F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}} = \frac{u^{n+\frac{1}{2}}_{i+1,j} + u^{n+\frac{1}{2}}_{i,j}}{2} + \frac{\Delta y}{4a}\left(p^n_{i+2,j} - 2p^n_{i+1,j} + p^n_{i,j}\right) - \frac{\Delta y}{4a}\left(p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}\right), \tag{2.45}$$

$$F^{n+\frac{1}{2}}_{v_{i,j+\frac{1}{2}}} = \frac{v^{n+\frac{1}{2}}_{i,j+1} + v^{n+\frac{1}{2}}_{i,j}}{2} + \frac{\Delta x}{4a}\left(p^n_{i,j+2} - 2p^n_{i,j+1} + p^n_{i,j}\right) - \frac{\Delta x}{4a}\left(p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}\right). \tag{2.46}$$

Reconsidering the pressure contribution for the horizontal fluxes computed with the new interpolation scheme we find by using (2.42) and adding the new term of the modified interpolation scheme,

$$\begin{aligned} F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2}}} &= \frac{u^{n+\frac{1}{2}}_{i+1,j} + u^{n+\frac{1}{2}}_{i,j}}{2} \\ &= \mathcal{G}_2 - \frac{1}{a}\frac{\Delta y}{4}\left(p^n_{i+2,j} + p^n_{i+1,j} - p^n_{i,j} - p^n_{i-1,j}\right) \\ &\quad + \frac{1}{a}\frac{\Delta y}{4}\left(p^n_{i+2,j} - 2p^n_{i+1,j} + p^n_{i,j}\right) - \frac{1}{a}\frac{\Delta y}{4}\left(p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}\right) \\ &= \mathcal{G}_2 - \frac{\Delta y}{a}\left(p^n_{i+1,j} - p^n_{i,j}\right). \end{aligned} \tag{2.47}$$

So the horizontal flux $F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}}$ depends on $p^n_{i+1,j}$ and $p^n_{i,j}$. Reconsidering the equation for the Laplacian of the pressure in (2.43) and substituting the intermediate fluxes using (2.47) (analogous to the vertical fluxes) we obtain:

$$\begin{aligned}\Delta\delta p^{n+1} =&\frac{2}{\Delta t}[\Delta y(F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}} - F^{n+\frac{1}{2}}_{u_{i-\frac{1}{2},j}}) + \Delta x(F^{n+\frac{1}{2}}_{v_{i,j+\frac{1}{2}}} - F^{n+\frac{1}{2}}_{v_{i,j-\frac{1}{2}}})]\\ =&\frac{2}{\Delta t}[\mathcal{G} - \frac{\Delta y}{a}\left(p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}\right) - \frac{\Delta x}{a}\left(p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}\right)].\end{aligned}$$

Therefore the pressure and the velocity are well coupled.

N.B. 2.4.5. *When we compare the linear interpolation of* (2.39) *with the modified linear interpolation of* (2.45)*, we notice that the added term corresponds to a third derivative in x-coordinate direction of the pressure multiplied by Δx*

$$\begin{aligned}&\frac{\Delta y}{4a}\left(p^n_{i+2,j} - 2p^n_{i+1,j} + p^n_{i,j}\right) - \frac{\Delta y}{4a}\left(p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}\right)\\ \approx&\frac{\Delta y}{4a}\Delta x^2\left(\partial^2_x\, p\right)^n_{i+1,j} - \frac{\Delta y}{4a}\Delta x^2\left(\partial^2_x\, p\right)^n_{i,j}\\ \approx&\frac{\Delta y}{4a}\,\frac{\Delta x\{\Delta x^2\left(\partial^2_x\, p\right)^n_{i+1,j} - \Delta x^2\left(\partial^2_x\, p\right)^n_{i,j}\}}{\Delta x}\\ \approx&\left(\frac{\Delta y}{a}\Delta x^2\right)\left(\frac{\Delta x}{4}\left(\partial^3_x p\right)^n_{i+\frac{1}{2},j} + \mathcal{O}\left(\Delta x^3\right)\right)\\ =&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\left(\frac{\Delta x}{4}\left(\partial^3_x p\right)^n_{i+\frac{1}{2},j} + \mathcal{O}\left(\Delta x^3\right)\right).\end{aligned}$$

Considering the difference $\left(F^{n+\frac{1}{2}}_{u_{i+\frac{1}{2},j}} - F^{n+\frac{1}{2}}_{u_{i-\frac{1}{2},j}}\right)$ *of the horizontal fluxes we have added the following term:*

$$\begin{aligned}&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\left(\frac{\Delta x}{4}\left(\partial^3_x p\right)^n_{i+\frac{1}{2},j} + \mathcal{O}\left(\Delta x^3\right)\right)\\ -&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\left(\frac{\Delta x}{4}\left(\partial^3_x p\right)^n_{i-\frac{1}{2},j} + \mathcal{O}\left(\Delta x^3\right)\right)\\ =&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\frac{\Delta x}{4}\left(\left(\partial^3_x\, p\right)^n_{i+\frac{1}{2},j} - \left(\partial^3_x\, p\right)^n_{i-\frac{1}{2},j}\right)\\ =&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\frac{\Delta x^2}{4}\left(\frac{\left(\partial^3_x\, p\right)^n_{i+\frac{1}{2},j} - \left(\partial^3_x\, p\right)^n_{i-\frac{1}{2},j}}{\Delta x}\right)\\ \approx&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\frac{\Delta x^2}{4}\left(\left(\partial^4_x\, p\right)^n_{i.j} + \mathcal{O}\left(\Delta x^4\right)\right)\\ =&\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\left(\frac{\Delta x^2}{4}\left(\partial^4_x\, p\right)^n_{i,j} + \mathcal{O}\left(\Delta x^4\right)\right).\end{aligned}$$

Following the same arguments for the vertical fluxes we have added at the right-hand side of the pressure equation (2.37) *the following small regularizing term,*

$$\left(\frac{1}{\Delta t} + \frac{\nu}{\Delta x^2} + \frac{\nu}{\Delta y^2}\right)^{-1}\left(\frac{\Delta x^2}{4}\left(\partial^4_x\, p\right)^n_{i,j} + \frac{\Delta y^2}{4}\left(\partial^4_y\, p\right)^n_{i,j} + \mathcal{O}\left(\Delta x^4\right) + \mathcal{O}\left(\Delta y^4\right)\right).$$

The method to prevent checkerboard oscillations is achieved by adding a small regularizing term which annihilates the spurious modes. This term does not disturb the consistency of our scheme but it increases its diffusivity.

2.5 Formulation of projection method in THETA

We will now introduce the projection method used in THETA (see [33]). We are given the Navier-Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\vec{\nabla})\vec{u} + \vec{\nabla}p - \vec{\nabla}\cdot\left[\nu\left(\vec{\nabla}\vec{u} + (\vec{\nabla}\vec{u})^T\right)\right] = \vec{f} \quad \text{in} \quad \Omega\times(0,T], \tag{2.48}$$

$$\vec{\nabla}\cdot\vec{u} = 0 \quad \text{in} \quad \Omega\times(0,T]. \tag{2.49}$$

We discretize the time derivative term with the BDF(2)-formula:

$$\frac{\partial \vec{u}}{\partial t} = \frac{3\vec{u}^{n+1} - 4\vec{u}^n + \vec{u}^{n-1}}{2\Delta t}$$

The projection scheme of THETA is written as follows:

Projection method of THETA

- **step 1**
  Seek $\vec{u}^*$ such that

$$\frac{3\vec{u}^* - 4\vec{u}^n + \vec{u}^{n-1}}{2\Delta t} + (\vec{u}^n\cdot\vec{\nabla})\vec{u}^* + \vec{\nabla}p^n - \vec{\nabla}\cdot\left[\nu\left(\vec{\nabla}\vec{u}^* + (\vec{\nabla}\vec{u}^*)^T\right)\right] = \vec{f}^{n+1}.$$

- **step 2**
  Seek pressure correction $\delta p^{n+1} = p^{n+1} - p^n$ such that

$$\Delta\delta p^{n+1} = \frac{3}{2\Delta t}\vec{\nabla}\cdot\vec{u}^*,$$
$$\vec{\nabla}\delta p^{n+1}\cdot\vec{n} = 0.$$

- **step 3**
  Compute the new pressure p^{n+1} from

$$p^{n+1} = p^n + \delta p^{n+1}.$$

- **step 4**
  Compute $\vec{u}^{n+1}$ such that

$$\vec{u}^{n+1} = \vec{u}^* - \frac{2\Delta t}{3}\vec{\nabla}\delta p^{n+1}.$$

.

N.B. 2.5.1. *There are some remarkable differences between the projection method used in THETA and the method by van Kan. Briefly summarized:*

- *The time discretization used by van Kan is an implicit Crank-Nicolson method whereas the time discretization in THETA uses a BDF(2)-formula. The latter scheme is A-stable.*

- *The nonlinear term* $(\vec{u} \cdot \vec{\nabla}\vec{u})$ *is treated fully explicit in the van Kan scheme. In THETA the term* $(\vec{u}^n \cdot \vec{\nabla})u^*$ *uses* $\vec{u}^n$ *from the previous time step for linearization. Due to the appearance of* $\vec{u}^*$ *this term is semi-implicit.*

- *The symmetric rate-of-strain tensor is treated implicitly in THETA as it is set to*

$$\vec{\nabla} \cdot \left[\nu \left(\vec{\nabla}\vec{u}^* + (\vec{\nabla}\vec{u}^*)^T \right) \right]$$

while the scheme of van Kan has a semi-implicit treatment since it is formulated as

$$-\nu\Delta \left(\frac{\vec{u}^{n+\frac{1}{2}} + \vec{u}^n}{2} \right)$$

for the Laplace term.

We have seen that the steps 1 and 2 of the projection method of van Kan can be transformed into a linear system of equations. In the DLR-THETA-code these equations are solved with a linear solver called biconjugate gradients stabilized (BiCGSTAB). BiCGSTAB is a variation of the conjugate gradients (CG) method [17]. The advantage is the convergence behavior which is much smoother than in the CG method. For further details please see Appendix A.

3. Sensors for identification of flow structures

Sensors for identification of flow structures are indicators for the length of flow structures (= eddies). It is important to know the structure of a flow in order to decide if a grid is fine enough to resolve the small scales of a flow. Thus, the information of a sensor can be useful to decide where to refine and where to coarsen the grid. In this chapter we will introduce a sensor for identification of flow structures, the Kármán-sensor. We will also demonstrate how the sensor can be computed numerically.

3.1 The von Kármán length scale

Inspired by the work of Rotta [23], Menter and Egorov [18] deduce an equation for the autocorrelation function for the turbulent fluctuations. From this equation the formula for the *von Kármán length scale* L_{vK} is derived (see Appendix B), given by

$$L_{vK} \;=\; \kappa \frac{S}{U''}$$

with $\kappa = 0.41$ and

$$\mathcal{S} \;=\; \sqrt{2\ \mathbb{S}(\vec{u}) : \mathbb{S}(\vec{u})},$$

$$U'' \;=\; \sqrt{\sum_{i=1}^{d} \sum_{j,k=1}^{d} \frac{\partial^2 u_i}{\partial x_j^2} \frac{\partial^2 u_i}{\partial x_k^2}}.$$

with the symmetric rate-of-strain tensor $\mathbb{S}(\vec{u})$ and " : " denotes the componentwise multiplication (compare to 2.2.1).
In 2D we obtain

$$S \;=\; \sqrt{2 \cdot (\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})^2 \;+\; 2 \cdot (\frac{\partial v}{\partial y})^2},$$

$$U'' \;=\; \sqrt{(\frac{\partial^2 u}{\partial x^2})^2 \;+\; 2 \cdot \frac{\partial^2 u}{\partial x^2} \cdot \frac{\partial^2 u}{\partial y^2} \;+\; (\frac{\partial^2 u}{\partial y^2})^2 \;+\; (\frac{\partial^2 v}{\partial x^2})^2 \;+\; 2 \cdot \frac{\partial^2 v}{\partial x^2} \cdot \frac{\partial^2 v}{\partial y^2} \;+\; (\frac{\partial^2 v}{\partial y^2})^2}.$$

In further description we will denote the von Kármán length scale as *Kármán-sensor*. Our aim is to analyze the quality of the sensor by applying it to several test cases. For the computation of the von Kármán-sensor we need the following:

- the velocity vector $\vec{u}$,
- the first partial derivatives $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$, $\frac{\partial v}{\partial y}$,
- the second order partial derivatives $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$, $\frac{\partial^2 v}{\partial x^2}$, $\frac{\partial^2 v}{\partial y^2}$.

From the solution of the flow solver THETA we obtain the velocity components of $\vec{u}$. Therefore we compute the first and second order partial derivatives from this solution. In the following sections we will show how to recover the first and second order partial derivatives numerically.

3.2 Computation of gradients

We will introduce two ways of computing gradients for unstructured meshes and one method for computing gradients on structured meshes. The quality of the methods, Green-Gauß formula, Least Square method and central difference scheme, is studied by applying to two simple test cases. We will, therefore, recover the gradients of sinus functions, in 1D and 2D and compute the maximal error by comparing the numerical to the analytical solution. At the end of the chapter we will compute the Kármán-sensor for a 2D flow field of a sinus type in order to see if the sensor is able to represent the small flow structures.

3.2.1 Computation of gradients on unstructured meshes using the Green-Gauß method

Unstructured meshes are important in solving fluid equations in a complex domain, Ω. Therefore we will introduce a method to compute the gradient

$$\vec{\nabla}\Phi = \begin{pmatrix} \frac{\partial \Phi}{\partial x} \\ \frac{\partial \Phi}{\partial y} \\ \frac{\partial \Phi}{\partial z} \end{pmatrix}$$

of a function $\Phi : \Omega \to \mathbb{R}$ at each mesh point. We will consider an unstructured mesh, Ω_P, with according dual mesh, Ω_D, with control volumina, Ω_{D_i}. From the integration theorem of Gauß we obtain the equation:

$$\int\limits_{\Omega_{D_i}} \vec{\nabla}\Phi(x,y,z)dV = \int\limits_{\partial\Omega_{D_i}} \Phi(x,y,z)\vec{n}dS,$$

where $\vec{n}$ denotes the outward pointing normal vector. The integral is approximated by:

$$\int\limits_{\Omega_{D_i}} \vec{\nabla}\Phi(x,y,z)dV \approx Vol(\Omega_{D_i}) \cdot \vec{\nabla}\Phi(x_i,y_i,z_i),$$

with $P_i = (x_i, y_i, z_i)$ the midpoint of the dual face Ω_{D_i}. Therefore we obtain the following formula for the gradient:

$$\vec{\nabla}\Phi(P_i) \approx \frac{1}{Vol(\Omega_{D_i})} \int\limits_{\partial\Omega_{D_i}} \Phi(x,y,z)\vec{n}dS.$$

Before we start to derive the algorithm of Green-Gauß gradient computation the following sketch illustrates the parts of the equation.

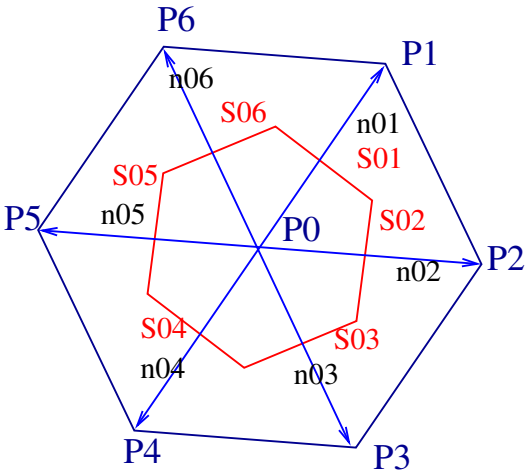

FIGURE 8: Primary mesh and its dual control volumina.

The vectors, which are denoted by $\vec{n}_{0i}$, point from P_0 to the neighboring points P_i. For the magnitude of the vector $\vec{n}_{0i}$ we choose the length of edge S_{0i} (considering the case in 2D). Hereby S_{0i} denotes the edge of the dual control volume, Ω_{D_0} belonging to point P_0, which meets with $\vec{n}_{0i}$. We explain examplarily how the integral,

$$\int\limits_{\partial\Omega_{D_0}} \Phi(x,y,z)\vec{n}dS$$

is approximated. The integral at the boundary of control volume Ω_{D_0} is approximated by

$$\int\limits_{\partial\Omega_{D_0}} \Phi(x,y,z)\vec{n}dS \approx \sum_{i=1}^{\#N(P_0)} \frac{1}{2}\left\{\Phi(P_0) + \Phi(P_i)\right\} \cdot \vec{n}_{0i}.$$

With $\#N(P_0)$ we denote the number of neighboring points of P_0. By $\frac{1}{2}\{\Phi(P_0) + \Phi(P_i)\}$ we compute the value of Φ at the midpoint of the edge $\{P_0, P_i\}$. We have to perform the computation on the whole grid for all nodes and obtain a simple version of the algorithm.

For all points P_i:

- Initialization $\vec{\nabla}\Phi(P_i) = 0$ for all i.
- Loop over all neighbors P_j and do:

$$\vec{\nabla}\Phi(P_i) \mathrel{+}= \frac{1}{Vol(\Omega_{D_i})} \cdot \frac{1}{2} \{\Phi(P_i) + \Phi(P_j)\} \cdot \vec{n_{ij}}$$

This simple version is point-based. We have two nested loops. An outer loop over all nodes and moreover, for each node, an inner loop over all neighbors. Hence, for each point the simple version takes into account all neighboring points in order to compute the contribution of the values of Φ at the midpoints of all edges e with $e = \{P_iP_j\}$. Therefore, all nodes are visited several times. We will improve the simple version of the algorithm. Our approach will be to use an edge based loop. The midpoint of each edge, $e = \{P_iP_j\}$ is taken into account in order to compute the contribution to the gradient of its start point P_i and to its end point P_j. Therefore the iteration loop is transferred over every edge. Hence the algorithm is more efficient. We should take note that the direction of the normal vector $\vec{n}_{ij}$ changes and therefore the sign switches.

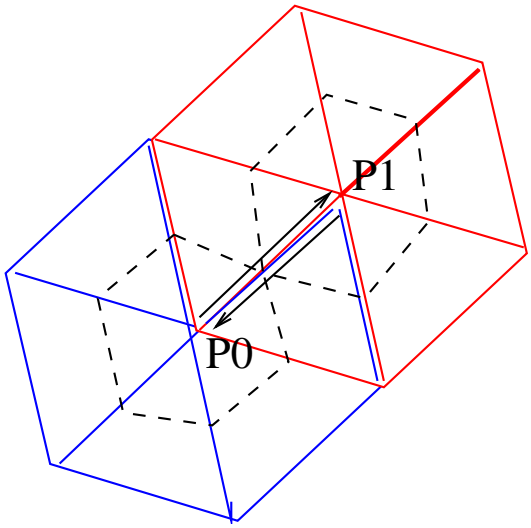

FIGURE 9: Sketch of two control volumina and its normal vectors.

We obtain the final version of the algorithm:

- Initialization: For each point P_i set $\vec{\nabla} \cdot \Phi(P_i) = 0$.
- For all edges e do:
  Set $P_i = e[0]$ the start point and $P_j = e[1]$ the end point of e.
- Add the edgewise contribution to the gradient of P_i and P_j

$$\vec{\nabla}\Phi[P_i] \mathrel{+}= \frac{1}{2} \{\Phi(P_i) + \Phi(P_j)\} \cdot \vec{n_{ij}},$$
$$\vec{\nabla}\Phi[P_j] \mathrel{-}= \frac{1}{2} \{\Phi(P_i) + \Phi(P_j)\} \cdot \vec{n_{ij}}.$$

In the following we will denote this algorithm as *Green-Gauß method*.

3.2.2 Computation of gradients on unstructured meshes using Least Square method

We can alternatively compute gradients on unstructured grids with a method based on the *Least Square method* [14], [9]. Starting point is the Taylor series of a function Φ around the point P_0

$$\Phi_i = \Phi_0 + \frac{\partial \Phi}{\partial x}(P_0)\ \Delta x_i + \frac{\partial \Phi}{\partial y}(P_0)\ \Delta y_i + \frac{\partial \Phi}{\partial z}(P_0)\ \Delta z_i + \mathcal{O}(h^2),$$

with $\Phi_0 = \Phi(P_0)$ and $\Phi_i = \Phi(P_i)$. We abort the Taylor series after the first derivatives. In vector notation we obtain

$$\Phi_i - \Phi_0 \ \approx\ (\Delta x_i,\ \Delta y_i,\ \Delta z_i\ ) \begin{pmatrix} \frac{\partial \Phi}{\partial x}(P_0) \\ \frac{\partial \Phi}{\partial y}(P_0) \\ \frac{\partial \Phi}{\partial z}(P_0) \end{pmatrix} \qquad i = 1, \ldots, n$$

where n is the number of neighbors. We will now consider all points $\{P_1, \ldots, P_n\}$ adjacent to P_0 and obtain the following linear equation system of the form $A\vec{x} = \vec{b}$

$$\begin{pmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \vdots & \vdots & \vdots \\ \Delta x_n & \delta y_n & \Delta z_n \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial \Phi}{\partial x}(P_0) \\ \frac{\partial \Phi}{\partial y}(P_0) \\ \frac{\partial \Phi}{\partial z}(P_0) \end{pmatrix} \ = \ \begin{pmatrix} \Phi_1 \ - \ \Phi_0 \\ \vdots \\ \Phi_n \ - \ \Phi_0 \end{pmatrix}.$$

The solution vector $\vec{x}$ provides the gradient $\frac{\partial \Phi}{\partial x_i}$. Matrix A contains all distances between P_0 and its neighbors P_i. Vector $\vec{b}$ contains the differences $\Phi_i - \Phi_0$ of the function values for Φ at P_0 and its neighboring points P_i. There are three components of the solution vector $\vec{x}$ which we are seeking. Three linear independent equations would be sufficient to determine $\vec{x}$ uniquely. The number of rows in A is equal to the number of neighboring points of P_0. In the general case grids with more than three neighbors for each point are considered. Thus, we can assume that we would obtain an *overdetermined system of equations* $A\vec{x} = \vec{b}$. Therefore a *Least Square* approach is applied because the determination of an exact solution is impossible. The aim is to approximate $\vec{x}$ by computing a $\vec{x}^*$ in order to minimize the error

$$||A\vec{x}^* \ - \ \vec{b}||.$$

Considering the Eucledian norm we obtain the classical *method of the smallest squares.*

3.2.3 Computation of gradients on structured meshes using central differences

On structured meshes the gradients can be computed by using a central difference formula [11]. The idea of central differences is to approximate the derivative $\frac{\partial u}{\partial x}$ with the difference quotient. Structured meshes are ideal because the cells are of hexahedral type and the nodes can be ordered, such that for each point P_i exists a right and a left neighbor P_{i-1} and P_{i+1}. Hence we can approximate the derivative by

$$(\frac{\partial u}{\partial x})_i \approx \frac{u_{i+1} - u_{i-1}}{2 \cdot \Delta x}, \tag{3.1}$$

where $(\frac{\partial u}{\partial x})_i$ denotes the first partial derivative at point P_i. For the second derivative we use the formula (3.1) twice and obtain,

$$(\frac{\partial^2 u}{\partial x^2})_i \approx \frac{u_{i+2} - 2 \cdot u_i + u_{i-2}}{(2 \cdot \Delta x)^2}.$$

An alternative relation for the second order derivative is based on the following theoretical approach to compute the first order derivative

$$(\frac{\partial u}{\partial x})_i \approx \frac{u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}}{\Delta x}, \tag{3.2}$$

where $u_{i-\frac{1}{2}}$ and $u_{i+\frac{1}{2}}$ denote virtual points which lie between node P_i and the neighboring nodes P_{i-1} and P_{i+1}. By using (3.2) twice we obtain the following formula for the second order derivatives

$$(\frac{\partial^2 u}{\partial x^2})_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}. \tag{3.3}$$

On a structured mesh the first and second order derivatives can easliy be computed with this method.

3.3 Numerical results for first order gradients

In this section we will present the numerical results we obtained by computing the gradients of the test cases "Sinus 1D" and "Sinus 2D" using the Green-Gauß formula, Least Square method and central differences. Note that in the present work only structured grids are used since turbulence simulations on unstructured meshes are not yet well established in literature.

3.3.1 Computation using Green-Gauß and Least Square method for test case "Sinus 1D"

We will study the quality of both methods for a simple test case in 1D. The function $u(x) = \sin(2\pi\omega x)$ is considered for $x \in \Omega = (0, 1)$ and $\omega \in \{4, 8, 16\}$. The first order derivative $u'(x) = 2\pi\omega \cos(2\pi\omega x)$ is recovered using Green-Gauß and Least Square method for several ω and mesh widths h at each point x. Therefore we use the DLR-THETA-code. We initiate the velocity field $\vec{u}$ with the reference function $u(x)$. By setting the computational time step size to zero and running the THETA-code, the initialization of the velocity field $\vec{u}$ takes place. The gradients are computed by calling the routines for the computation with Green-Gauß and Least Square method with input vector $\vec{u}$. Using the solution we compute the maximal error and set it into relation with mesh width h and ω. The results are shown in the following graphs.

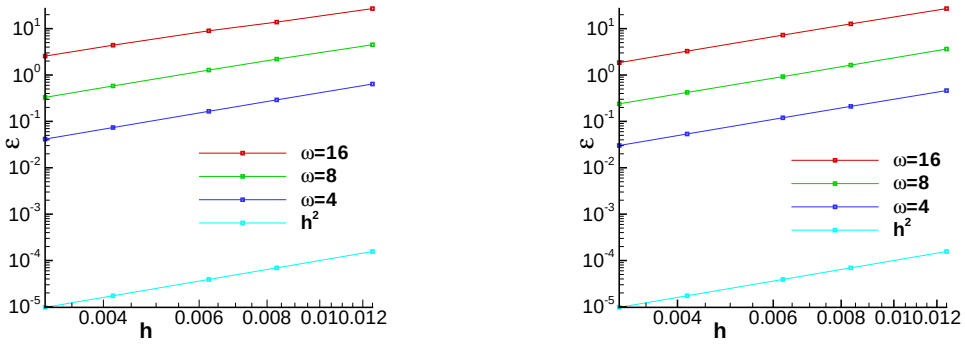

FIGURE 10: Maximal error ϵ for $u'(x)$ using Green-Gauß (left) and Least Square method (right) for test case "Sinus 1D" for various values for mesh width h and ω.

We will take into account the second test case before we start to analyze the results of the computations.

3.3.2 Computation using Green-Gauß and Least Square method for test case "Sinus 2D"

We will test the computation of the gradient $\frac{\partial u}{\partial x}$ with the test case "Sinus 2D" using Green-Gauß and Least Square method. We will consider the reference function

$$\vec{u}(x,y,t) = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} \sin(t) \tag{3.4}$$

with

$$\begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = \begin{pmatrix} \pi \sin^2(\omega\pi x)\sin(2\omega\pi y) \\ -\pi \sin(2\omega\pi x)\sin^2(\omega\pi y) \end{pmatrix}. \tag{3.5}$$

Together with the pressure

$$p(x,y,t) = \cos(\omega\pi x)\sin(\omega\pi y)\sin t$$

the velocity field $\vec{u}(x,y,t)$ fulfills the Navier-Stokes equations. Note that this function is time-dependent. We will consider the partial derivative of u in x-direction which reads thus:

$$\frac{\partial u}{\partial x} = 2\pi^2\omega \sin(\omega\pi x)\cos(\omega\pi x)\sin(2\omega\pi y). \tag{3.6}$$

As in the preceding subsection we use the THETA-code for computation of the gradients. Hereby we initiate the velocity field $\vec{u}$ by using the reference function $\vec{u}(x,y,t)$ and setting $t = 0.01$. The maximal error for various mesh widths h and ω is shown in the following graphs:

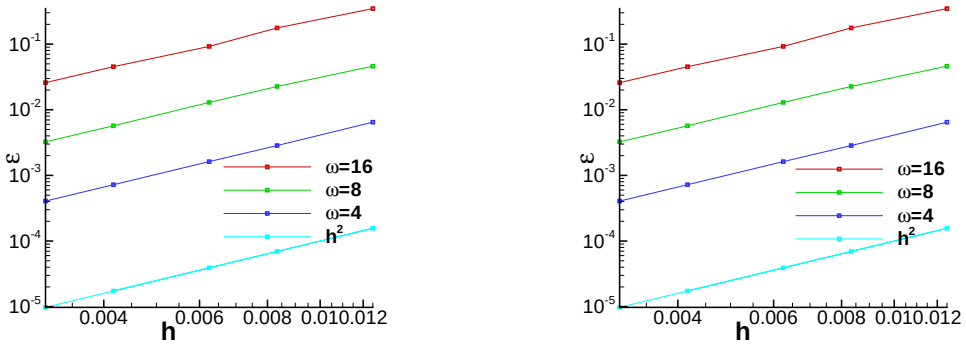

FIGURE 11: Maximal error ϵ for $\frac{\partial u}{\partial x}$ using Green-Gauß (left) and Least Square method (right) for test case "Sinus 2D" for various mesh widths h and values for ω.

We consider the results of 10 and 11. By comparing the recovered derivatives u_i' with the exact solution $u'(x_i)$ with $h = |x_{i+1} - x_i|$ for $i = 1, \ldots, N$ we obtain

$$\max_{i=1,\ldots,N} |u_i' - u'(x_i)| \leq const \cdot \mathcal{O}(h^2).$$

Therefore we can state the error of the computation using Green-Gauß and Least Square method is of quadratic order. Considering the results for test case "Sinus 1D" (Fig. 10) we see that the computation using Least Square method is slighty better than the computation using Green-Gauß method. For test case "Sinus 2D" (see Fig. 11) the quality of both computations is almost the same.

3.3.3 Computation using central differences for test cases "Sinus 1D" and "Sinus 2D"

We will consider the test case "Sinus 1D" as in 3.3.1 and "Sinus 2D" as in 3.3.2 using the central difference formula (3.3) to recover the first order partial derivatives. The first order derivative is computed using a C-program called *interpolate.c* which reads the mesh points out of a grid. The values of the function are computed at each point of the mesh by using the reference functions. In the next step the first order derivative is computed at each mesh node. By comparing the results to the analytical solution of the first order derivative we compute the maximal error for various mesh widths h and ω. The results are illustrated in the following graphs.

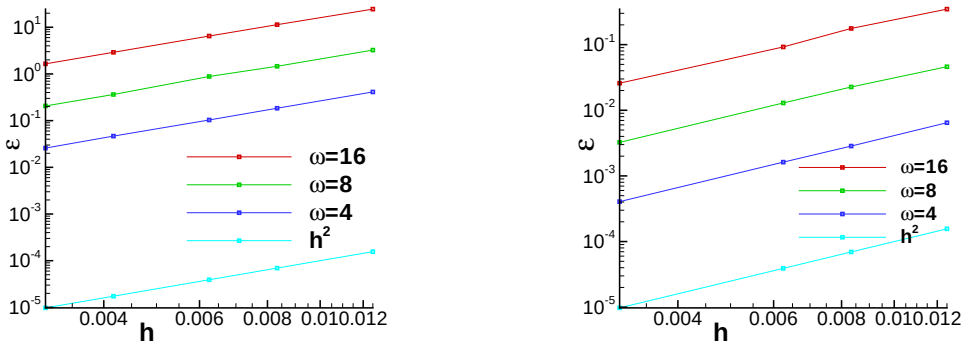

FIGURE 12: Maximal error ϵ of first order partial derivatives for test case "Sinus 1D" (left) and "Sinus 2D" (right) for computation using central differences.

Again, by comparing the recovered derivatives, u'_i, with the exact solution, $u'(x_i)$, second order accuracy can be clearly seen. For test case "Sinus 1D" the computation using central differences (Fig. 12 (left)) is slightly better than the computation using Green-Gauß method and as good as the computation using Least Square method (Fig. 10). For test case "Sinus 2D" (Fig 12(right), 11) the error of the three methods is almost the same.

3.4 Numerical results for second order gradients

In this section we will consider the computation of the second order gradients using Green-Gauß formula, Least Square method and central differences.

3.4.1 Computation using Green-Gauß and Least Square method for test case "Sinus1D"

We will consider the test case "Sinus 1D" as in 3.3.1. As before, we use the DLR-THETA-code for computing the second order derivative $u''(x) = -4\pi^2\omega^2 \sin(2\pi\omega x)$. The THETA-code first computes the first order derivative $u'(x)$ using Green-Gauß or Least Square method. Subsequently the second order derivatives are computed from the first order gradients. Hereby a Green-Gauß-like scheme is used. So far the computation of the second order derivatives with the Least Square method is not implemented in the THETA-code. The results of the maximal error are shown in the following graphs.

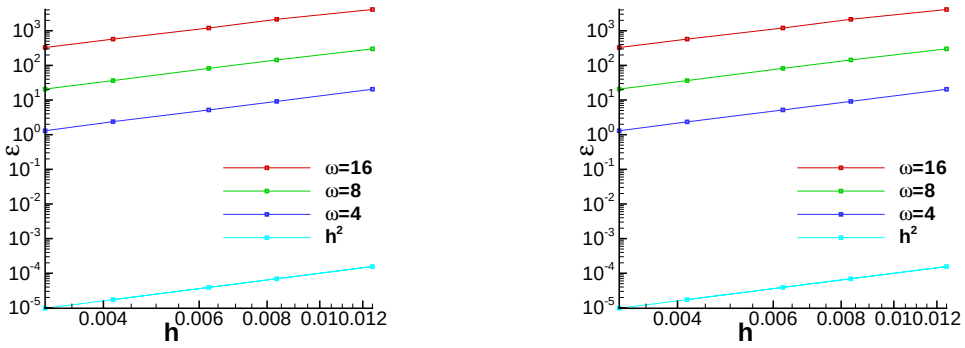

FIGURE 13: Maximal error for u'' for computation of u' with Green-Gauß (left) and Least Square (right) for test case "Sinus 1D".

The analysis of the results follows after the second test case.

3.4.2 Computation using Green-Gauß and Least Square method for test case "Sinus 2D"

We will consider the test case "Sinus 2D" as in 3.3.2. We compute the Laplacian $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ of the reference function using the THETA-code. The second order partial derivatives of u are

$$\frac{\partial^2 u}{\partial x^2} = 2\pi^3\omega^2 \left[\cos^2(\omega\pi x) - \sin^2(\omega\pi x)\right] \sin(2\omega\pi y),$$
$$\frac{\partial^2 u}{\partial y^2} = -4\pi^3\omega^2 \sin^2(\omega\pi y) \sin(2\omega\pi y).$$

As before, the first order partial derivatives $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$ are computed using Green-Gauß or Least Square method. Then the second order derivatives are computed from the first order gradients using a Green-Gauß-like scheme. The results of the maximal error is shown in the following figures.

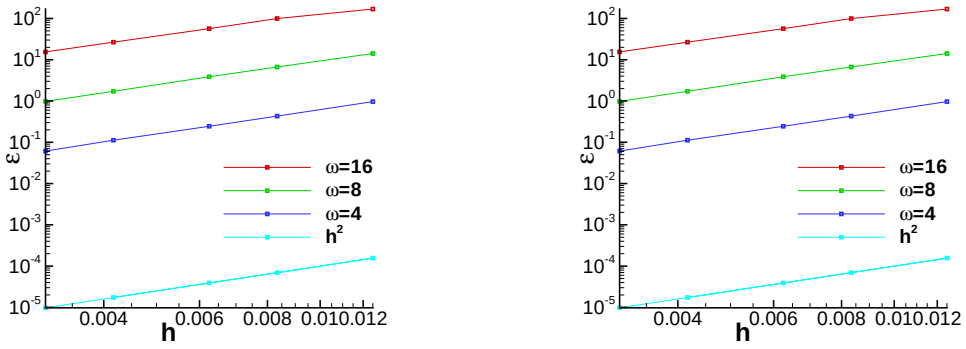

FIGURE 14: Maximal error for Δu for computation of $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$ with Green-Gauß (left) and Least Square (right) for test case "Sinus 2D".

The error of the computation of the second order gradients for test cases "Sinus 1D" (Fig. 13) and "Sinus 2D" (Fig. 14) has the order $\mathcal{O}(h^2)$. The quality of the computation using Green-Gauß and Least Square method is almost identical.

3.4.3 Computation using central differencs for test cases "Sinus 1D" and "Sinus 2D"

We will apply the central differences method to the test cases "Sinus 1D" 3.3.1 and "Sinus 2D" (see 3.3.2). The second order derivative u'' of "Sinus 1D" and the Laplacian Δu of "Sinus 2D" are computed using the formula (3.3) and the C-program *interpolate.c*. The results of the maximal error are shown in the following graphs.

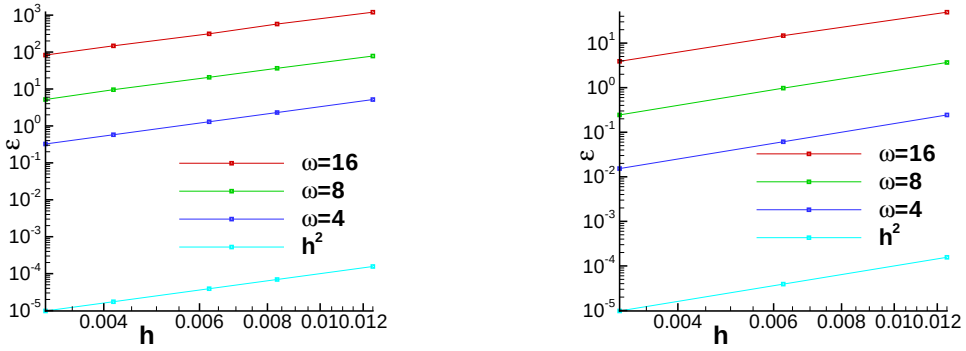

FIGURE 15: Maximal error for second order partial derivatives for test case "Sinus 1D" (left) and "Sinus 2D" (right) for computation using central differences.

We can see that the error has the order $\mathcal{O}(h^2)$. By comparing the results of test case "Sinus 1D" and "Sinus 2D" using central differences (Fig. 15) to the results obtained by the computation using Green-Gauß and Least Square method (Fig. 13, 14) we

observe that the error for central differences is smaller (by a factor 4). The reason for this is that according to formula (3.3) and (3.2) the central differences are computed on a smaller stencil which has half the width of the mesh spacing and has, therefore, half the width of the stencil for the computation with Green-Gauß and Least Square method. As the error has the order $\mathcal{O}(h^2)$ the result is a factor 4.

N.B. 3.4.1. *The methods we used for recovering first and second order gradients have an error which is of order $\mathcal{O}(h^2)$ as we can see in the preceding figures. Consequently, Green-Gauß and Least Square method are sufficiently good for recovering gradients on structured meshes in THETA. The computation using Least Square method needs more computational time than the Green-Gauß method. In addition, the Least Square method is not implemented in THETA for computing the second order derivatives. Therefore, if not stated otherwise, we will use Green-Gauß method for further computations.*

3.5 Numerical results of Kármán-sensor for test case "Sinus 2D"

In this section we will compute the Kármán-sensor for the test case "Sinus 2D" (see 3.3.2). The Kármán-sensor has been designed for fully developed turbulent flow and for the corresponding turbulent structures which are typically much smaller than the large flow structures. Nevertheless, it might be instructive to consider the Kármán length scale for an academic large scale flow structure.

By applying to this test case we want to see if the sensor is able to detect the structures of a flow field. We will compute the solution of the velocity field, $\vec{u}$, with THETA using initial conditions $\vec{u} = 0$ and a model file which computes the fluxes using the analytical solution in each time step. In the first step we want to analyze the flow structure of this sinus type function. Therefore the velocity in x- and y-coordinate directions (u and v) are plotted in Figure 16.

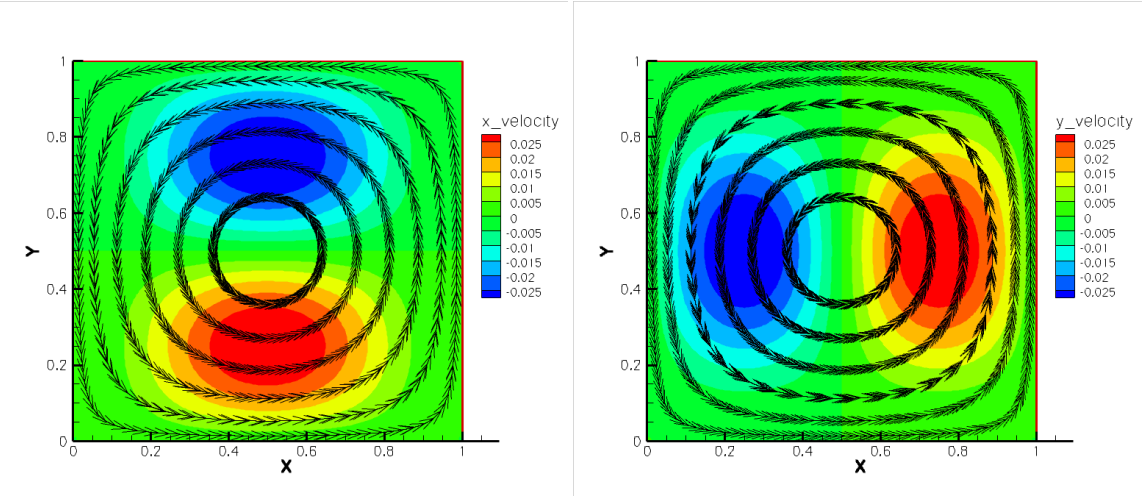

FIGURE 16: Velocities u (left) and v (right) for $\omega = 1$ and streamtraces $\vec{u} = (u, v)$.

From Figure 16, particularly from the streamtraces, we can conclude that for $\omega = 1$ one vortex exists in the domain. In the following we will consider only the velocity

component in x-coordinate direction.

The first question we need to ask is how to define a characteristic "diameter of a vortex" for the present flow. A proposal is sketched in Figure 17.

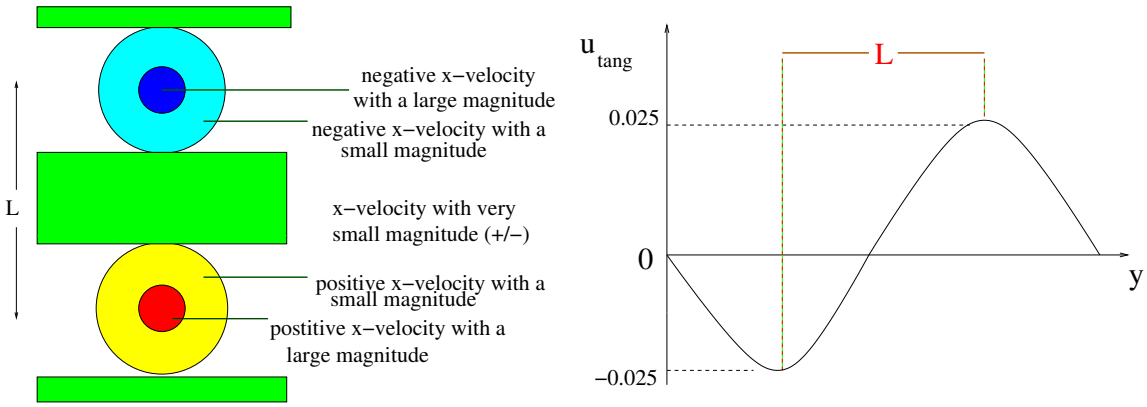

FIGURE 17: Schematic of the tangential velocity (left) and its corresponding length (right).

A simple characteristic length (or "characteristic diameter" of the vortex) is given by the distance (in y-coordinate direction) between maximum x-velocity $+u_{tang}$ and minimum x-velocity $-u_{tang}$.

The second question we need to ask is which value of L_{vK} corresponds to this vortex diameter L. In order to answer this question we will compute the Kármán-sensor (3.1) by using a Least Square method for the first order derivatives and Green-Gauß method for the second order derivatives (according to 3.2). In the following figures 18 and 19 the Kármán-sensor is shown for various values for ω.

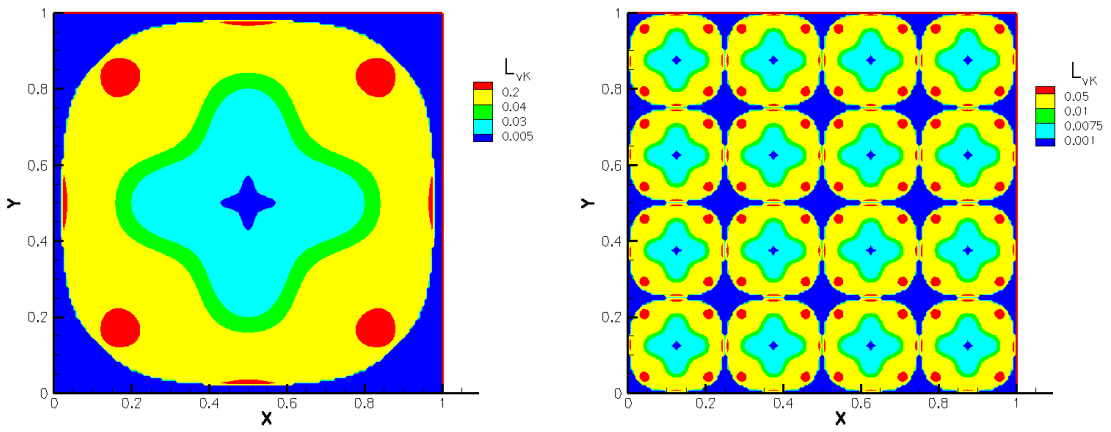

FIGURE 18: Kármán-sensor for $\omega = 1$ (left) and $\omega = 4$ (right).

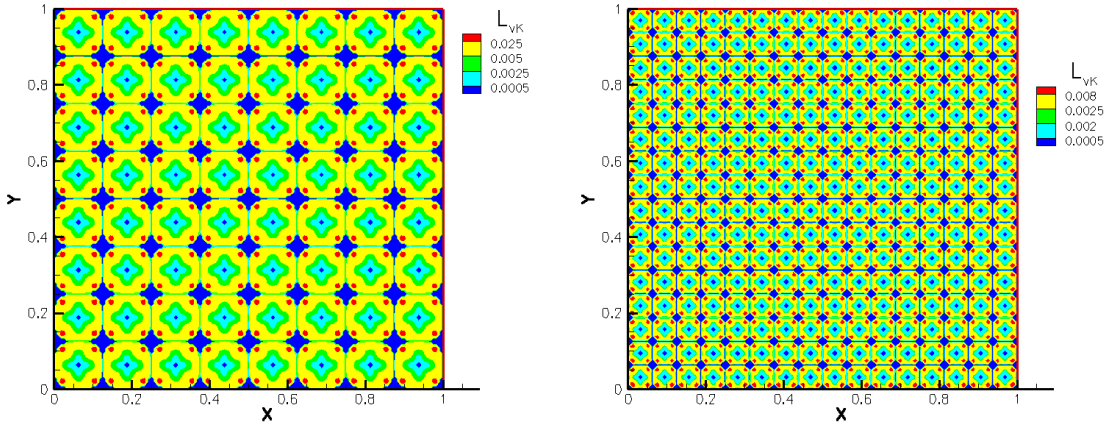

FIGURE 19: Kármán-sensor for $\omega = 8$ (left) and $\omega = 16$ (right).

We will consider $\omega = 1$ and discuss the following three regions in the characteristic pattern of L_{vK} in order to answer the second question.

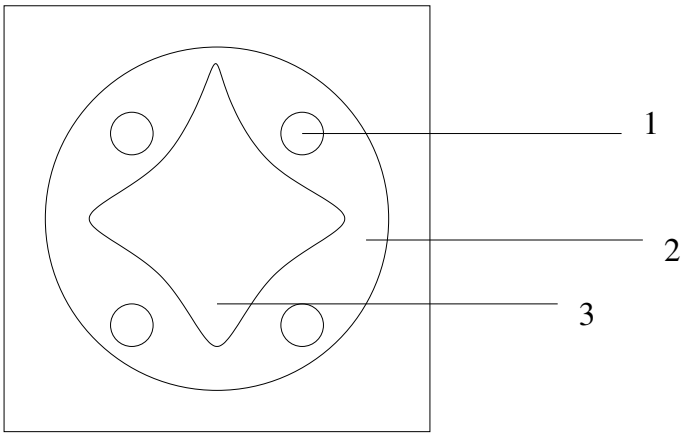

FIGURE 20: Three regions of the Kármán-sensor.

- region 1: Outer region where $|u_{tang}|$ is decaying in radial direction. L_{vK} takes very large values. Therefore it is not suitable.
- region 2: Region where $|u_{tang}|$ takes maximum values. L_{vK} takes values between 0.1 and 0.2. Therefore it is the most suitable region.
- region 3: Inner region of the vortex where $|u_{tang}|$ is growing in radial direction. L_{vK} takes very small values. Therefore it is not suitable.

Denote L the diameter of the vortical structure where u_{tang} is largest. For $\omega = 1$ $L \approx 0.5m$. For $\omega = 4, 8, 16$ one would expect a characteristic vortex diameter of $0.125,\ 0.0625,\ 0.03125$. For the Kármán-sensor we obtain in region 2

- $0.04 \lessapprox L_{vK} \lessapprox 0.2 \qquad \text{for } \omega = 1 \quad \longrightarrow \quad L_{vK} \approx 0.12,$
- $0.01 \lessapprox L_{vK} \lessapprox 0.05 \qquad \text{for } \omega = 4 \quad \longrightarrow \quad L_{vK} \approx 0.03,$

- $0.005 \lessapprox L_{vK} \lessapprox 0.025$ for $\omega = 8 \longrightarrow L_{vK} \approx 0.015$,
- $0.0025 \lessapprox L_{vK} \lessapprox 0.008$ for $\omega = 16 \longrightarrow L_{vK} \approx 0.005$.

Relative changes in vortex diameter can be detected reasonably well. Regarding the absolute value of L_{vK} it is smaller by approximately a factor 4 compared to the diameter found taking the distance between $+u_{tang}$ and $-u_{tang}$.

N.B. 3.5.1. *The formula for the Kármán-sensor contains the second order gradients in the denominator. The second order derivative of the test function "Sinus 2D" becomes zero in the interval* $[0, 1]$*. Therefore, to avoid a division by zero, we inserted a constraint into the routine such that if* $U'' < 0.1$ *then set* $U'' = 0.1$.

4. Simulation of turbulent channel flow using LES

Our goal is to study the behavior of the Kármán-sensor for identification of turbulent structures. Therefore, test cases of turbulent flows like the channel flow are necessary. Turbulent channel flow is a standard test case for wall bounded flows. In this chapter we will introduce the main idea of the large-eddy simulation and continue with a general description of the test case.

4.1 Large-eddy simulation

In large-eddy simulation (LES), the larger, unsteady turbulent motions are directly represented, the effects of the smaller scale motions are modeled. The cost of LES lies between RANS and DNS. As the large-scale unsteady motions are represented explicitly, LES can be expected to be more accurate and reliable than Reynolds-stress models. There are four conceptual steps in LES (see [21, p.558]):

- A *filtering* operation is defined to decompose the velocity $\vec{u}$ into a resolved component $\overline{\mathbf{u}}$ and into a residual (or subgrid-scale, SGS) component $\mathbf{u}'$. The filtered velocity field $\overline{\mathbf{u}}$ represents the motion of the large eddies, whereas the effect of the smaller scales $\mathbf{u}'$ has to be modeled. Note that we write $\overline{\mathbf{u}}$ instead of $\overline{\vec{u}}$.
- The equations for the evolution of the filtered velocity field are derived from the Navier-Stokes equations. Due to the filter properties the equations are transformed into the standard form, with the momentum equation containing the Reynolds-stress tensor.
- Closure is obtained by modeling the Reynolds-stress tensor, e.g., by using an eddy-viscosity model.
- The filtered scales, $\overline{\mathbf{u}}$, are solved numerically. They provide an approximation to the large-scale motions. The effect of the smaller scales is modeled using a proper model for the Reynolds-stress tensor.

There are different LES turbulence models. We will now introduce the model used in our test case.

4.1.1 Filtering

Filtering is used to separate the scales into filtered scales and residual components or subgrid scales (SGS). In general, a scale separation operator $\mathcal{T}$ is applied to $\vec{u}(\vec{x},t)$ with time t and space coordinate $\vec{x}$. The following decomposition is obtained:

$$\vec{u}(\vec{x},t) = \overline{\mathbf{u}} + \mathbf{u}'$$

where $\overline{\mathbf{u}} = \mathcal{T}(\vec{u})$ is the resolved part of $\vec{u}$ and $\mathbf{u}'$ denotes the unresolved part. Similarly, the scale separation operator is applied to the pressure, p. We will require that the scale separation operator have some important properties, i.e. the following assumptions should be valid:

- Linearity: $\overline{c\mathbf{u} + \mathbf{v}} = c\overline{\mathbf{u}} + \overline{\mathbf{v}}, \quad c \in \mathbb{R}$.
- Commutativity with differentiation in space: $\overline{(\frac{\partial \mathbf{u}}{\partial x_j})} = \frac{\partial}{\partial x_j}\overline{\mathbf{u}}$.
- Commutativity with differentiation in time: $\overline{(\frac{\partial \mathbf{u}}{\partial t})} = \frac{\partial}{\partial t}\overline{\mathbf{u}}$.
- Projection identity : $\overline{\overline{\mathbf{u}}} = \overline{\mathbf{u}}$.

In our computation the dual grid decomposes the velocity field $\vec{u}$ into large scales which can be resolved by the grid whereas the small scales cannot be resolved. Consequently, they have to be modeled. The following sketch shows the filtering process via a dual grid.

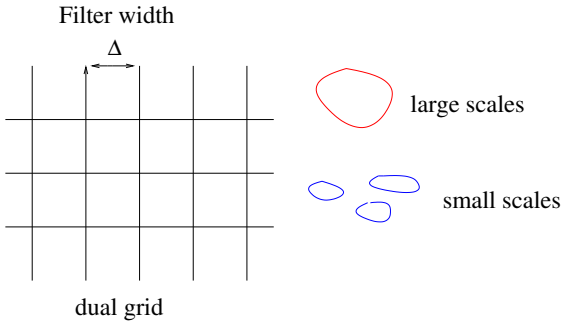

FIGURE 21: The dual grid as a filter with filter width Δ.

The filtered Navier-Stokes equations read

$$\partial_t \overline{\mathbf{u}} + \vec{\nabla} \cdot (\overline{\mathbf{u} \otimes \mathbf{u}}) + \vec{\nabla}\overline{p} - \vec{\nabla} \cdot (2\nu \mathbb{S}(\overline{\mathbf{u}})) = 0,$$
$$\vec{\nabla} \cdot \overline{\mathbf{u}} = 0,$$

where $\overline{\mathbf{u}}$ and $\overline{p}$ denote the filtered quantities of velocity field $\vec{u}$ and pressure p. The Reynolds-stress tensor is defined as

$$\tau = \overline{\mathbf{u} \otimes \mathbf{u}} - (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}})$$

With the Reynolds-stress tensor we can rewrite the Navier-Stokes equations

$$\partial_t \overline{\mathbf{u}} + \vec{\nabla} \cdot (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) + \vec{\nabla} \cdot \tau + \vec{\nabla} \overline{p} - \vec{\nabla} \cdot (2\nu \mathbb{S}(\overline{\mathbf{u}})) = 0, \tag{4.1}$$

$$\vec{\nabla} \cdot \overline{\mathbf{u}} = 0. \tag{4.2}$$

For a general three-dimensional flow we have four unknowns from the velocity field, $\vec{u}$ and pressure, p. The four equations from (4.1) and(4.2) contain more than four unknowns as the Reynolds-stress tensor, τ, contains six independent components. Consequently, the problem is unclosed. For this reason the Reynolds-stress tensor has to be modeled.

4.1.2 WALE - model

In this thesis we will use the so-called wall-adapting local eddy-viscosity (WALE) model [12] to emulate the Reynolds-stress tensor. The linear eddy-viscosity model

$$\tau = -2\nu_t \mathbb{S}(\overline{\mathbf{u}}) \tag{4.3}$$

is used to relate the residual stress to the filtered rate-of-strain. The effective viscosity is defined as

$$\nu_e = \nu + \nu_t. \tag{4.4}$$

From (4.3) the equations can be rewritten as

$$\partial_t \overline{\mathbf{u}} + \vec{\nabla} \cdot (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) - \vec{\nabla} \cdot (2\nu_t \mathbb{S}(\overline{\mathbf{u}})) + \vec{\nabla} \overline{p} - \vec{\nabla} \cdot (2\nu \mathbb{S}(\overline{\mathbf{u}})) = 0, \tag{4.5}$$

$$\vec{\nabla} \cdot \overline{\mathbf{u}} = 0.$$

With (4.4) the equation (4.5) can be transformed into

$$\partial_t \overline{\mathbf{u}} + \vec{\nabla} \cdot (\overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) - \vec{\nabla} \cdot (2\nu_e \mathbb{S}(\overline{\mathbf{u}})) + \nabla \overline{p} = 0.$$

In the WALE model the eddy viscosity ν_t is modeled by

$$\nu_t = C_w \Delta^2 \frac{\sum_{i,j=1}^{3} (S_{ij}^d S_{ij}^d)^{\frac{3}{2}}}{\sum_{i,j=1}^{3} (S_{ij} S_{ij})^{\frac{5}{2}} + \sum_{i,j=1}^{3} (S_{ij}^d S_{ij}^d)^{\frac{5}{4}}}$$

with $S_{ij} = (\mathbb{S}(\overline{\mathbf{u}}))_{ij}$ and S_{ij}^d defined by

$$S_{ij}^d = \frac{1}{2}(g_{ij}^2 + g_{ji}^2) - \frac{1}{3}\delta_{ij} g_{kk}^2, \qquad g_{ij}^2 = g_{ik} g_{kj}, \qquad g_{ij} = \frac{\partial \overline{\mathbf{u}}_i}{\partial x_j}$$

with WALE constant $C_w = 0.1$.

4.2 Turbulent channel flow

Description of the flow

A channel is a rectangular duct of length $L = 2\pi m$ and height $2H = 2m$. It is sufficiently long ($L \gg H$) and has a large aspect ratio ($b = \pi m \gg H$). The following

sketch demonstrates such a channel.

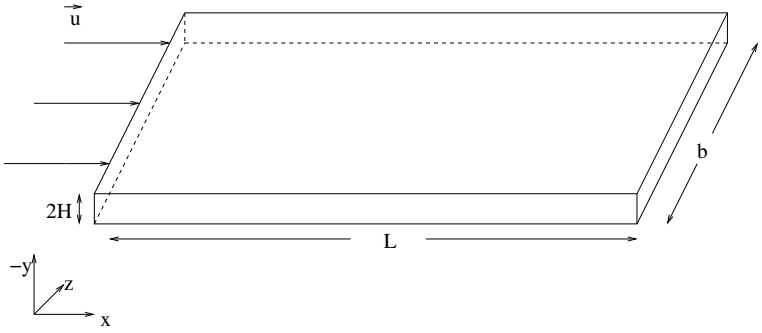

FIGURE 22: Sketch of the channel.

x represents streamwise direction; y wall normal direction; and z spanwise direction. For the boundaries ideally both in streamwise and spanwise direction periodic boundary conditions should be used. However, in the THETA-code it is only possible to use periodic boundary conditions in one direction. Therefore, we will use symmetry boundary condition in spanwise direction. To model an infinitely long channel we choose periodic boundary conditions in streamwise direction. At the viscous upper and lower wall, the no-slip boundary condition is prescribed such that the velocity field vanishes, i.e. $\vec{u} = (u, v, w) = \vec{0}$.

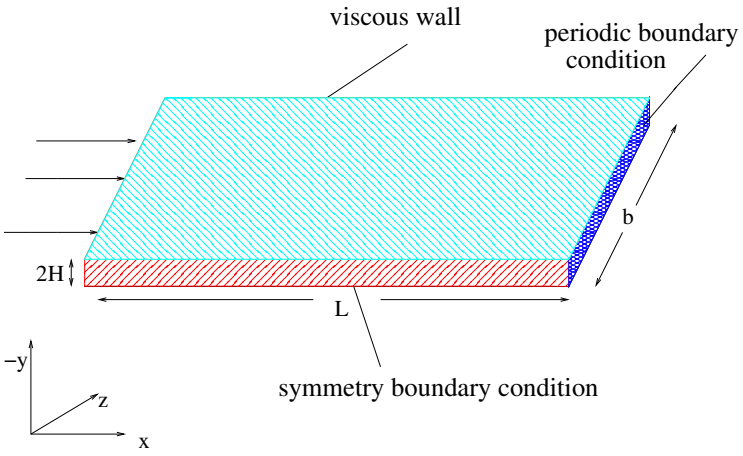

FIGURE 23: Boundary conditions for channel flow.

Model for the channel flow

We start with the Reynolds-averaged Navier-Stokes equations *ansatz* (see [33, p.42])

$$\partial_t U + \vec{\nabla} \cdot (\vec{U} \otimes \vec{U}) + \vec{\nabla} \cdot \langle \vec{u}' \otimes \vec{u}' \rangle + \frac{1}{\rho} \vec{\nabla} P - \vec{\nabla} \cdot (2\nu \mathbb{S}(\vec{U})) = 0, \tag{4.6}$$

$$\vec{\nabla} \cdot \vec{U} = 0, \tag{4.7}$$

with $\vec{U} = (U, V, W) = \langle \vec{u} \rangle$ the mean velocity field, $\vec{u}' = (u', v', w')$ the fluctuation and $P = \langle p \rangle$ the mean pressure with ensemble averaging filter $\langle \cdot \rangle$. We will describe the averaging filter and the fluctuation part in Section 4.6. Since the width b of

the channel is large compared with H the mean flow is independent of z, i.e. all quantities associated with z are zero. It follows that $W = 0$ and $\partial(\cdot)/\partial z = 0$. In componentwise notation we obtain:

$$\begin{aligned} U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + \frac{\partial}{\partial x}\langle u'u'\rangle + \frac{\partial}{\partial y}\langle u'v'\rangle + \frac{1}{\rho}\frac{\partial P}{\partial x} - \nu\frac{\partial^2 U}{\partial x^2} - \nu\frac{\partial^2 U}{\partial y^2} = 0, \\ U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + \frac{\partial}{\partial x}\langle u'v'\rangle + \frac{\partial}{\partial y}\langle v'v'\rangle + \frac{1}{\rho}\frac{\partial P}{\partial y} - \nu\frac{\partial^2 V}{\partial x^2} - \nu\frac{\partial^2 V}{\partial y^2} = 0, \\ \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0. \end{aligned}$$

In the *fully developed* region (large x), the mean velocity no longer changes with x. The mean flow is considered one-dimensional with mean velocity only depending on y. Hence $\partial(\cdot)/\partial x = 0$. From this we obtain:

$$\begin{aligned} V\frac{\partial U}{\partial y} + \frac{\partial}{\partial y}\langle u'v'\rangle \; + \; \frac{1}{\rho}\frac{\partial P}{\partial x} - \nu\frac{\partial^2 U}{\partial y^2} = 0, \\ V\frac{\partial V}{\partial y} + \frac{\partial}{\partial y}\langle v'v'\rangle \; + \; \frac{1}{\rho}\frac{\partial P}{\partial y} - \nu\frac{\partial^2 V}{\partial y^2} = 0, \\ \frac{\partial V}{\partial y} = 0. \end{aligned}$$

Integration of the continuity equation $\frac{\partial V}{\partial y} = 0$ from $y = 0$ to $y = 2H$, using the boundary condition $V = 0$ at $y = 0$ and $V = 0$ at $y = 2H$, leads to $V \equiv 0$. As the derivatives in x- and z-coordinate direction vanish, we can conclude that the solution for $\vec{u}$ is thus:

$$\vec{u} = u(y)\vec{e_1}.$$

So the mean flow is parallel to the x-axis and only depends on y. Consequently, we have two homogeneous directions, in x- and in z-direction

$$\frac{\partial}{\partial y}\langle u'v'\rangle \; + \; \frac{1}{\rho}\frac{\partial P}{\partial x} - \nu\frac{\partial^2 U}{\partial y^2} = 0, \tag{4.8}$$

$$\frac{\partial}{\partial y}\langle v'v'\rangle \; + \; \frac{1}{\rho}\frac{\partial P}{\partial y} = 0. \tag{4.9}$$

Equation (4.9) can be integrated to give:

$$\begin{aligned} & \int_{y'=0}^{y} \left(\frac{\partial}{\partial y}\langle v'^2\rangle + \frac{1}{\rho}\frac{\partial P}{\partial y} \right) dy' = 0 \\ \Leftrightarrow \qquad & \langle v'^2\rangle + \frac{P}{\rho} - \langle v'^2\rangle|_{y=0} - \frac{P}{\rho}|_{y=0} = 0 \\ \Leftrightarrow \qquad & \langle v'^2\rangle + \frac{P}{\rho} = \frac{P}{\rho}|_{y=0} \end{aligned}$$

as v' is zero at the wall $y = 0$. By taking the derivative in streamwise direction on both sides of this equation, we obtain

$$\frac{\partial}{\partial x}\left(\langle v'^2\rangle + \frac{P}{\rho} \right) = \frac{1}{\rho}\frac{\partial P|_{y=0}}{\partial x}.$$

This leads to:

$$\frac{1}{\rho}\frac{\partial P}{\partial x} = \frac{1}{\rho}\frac{\partial P|_{y=0}}{\partial x} \tag{4.10}$$

as the derivative in x-direction of $\langle v' \rangle^2$ vanishes. Equation (4.10) shows that the mean pressure gradient in streamwise direction is uniform in wall normal direction (as it is equal to the pressure gradient at $y = 0$), i.e. independent of y.

Insertion into the momentum equation for U (4.8) gives:

$$\frac{\partial}{\partial y}\left(\langle u'v' \rangle - \nu\frac{\partial U}{\partial y}\right) = -\frac{1}{\rho}\frac{\partial P|_{y=0}}{\partial x}. \tag{4.11}$$

The term on the right-hand side only depends on x whereas the term on the left-hand side only depends on y, as the velocity field only depends on it. From this we can conclude that both must be constant. Hence, by integration of equation (4.11) from $y = 0$ to $y = 2H$ we obtain:

$$\begin{aligned}
&\int_{y=0}^{y=2H} \frac{\partial}{\partial y}\left(\langle u'v' \rangle - \nu\frac{\partial U}{\partial y}\right) dy \\
&= \langle u'v' \rangle|_{y=2H} - \nu\frac{\partial U}{\partial y}|_{y=2H} - \langle u'v' \rangle|_{y=0} + \nu\frac{\partial U}{\partial y}|_{y=0} \\
&= -\frac{2H}{\rho}\frac{\partial P}{\partial x}.
\end{aligned}$$

The no-slip boundary condition $u'|_{y=0} = v'|_{y=0} = u'|_{y=2H} = v'|_{y=2H} = 0$ implies that $\langle u'v' \rangle|_{y=0} = \langle u'v' \rangle|_{y=2H} = 0$. So we obtain the equation:

$$-\nu\frac{\partial U}{\partial y}|_{y=2H} + \nu\frac{\partial U}{\partial y}|_{y=0} = -\frac{2H}{\rho}\frac{\partial P}{\partial x}. \tag{4.12}$$

We will introduce wall shear stress τ_w and friction velocity u_τ

$$\tau_w = \rho\nu\frac{\partial U}{\partial y}|_{y=0}, \qquad u_\tau = \sqrt{\frac{|\tau_w|}{\rho}}. \tag{4.13}$$

From equation (4.12) we obtain the relation

$$2\frac{\tau_w}{\rho} = -\frac{2H}{\rho}\frac{\partial P}{\partial x}.$$

Therefore we obtain for the pressure gradient

$$\frac{\partial P}{\partial x} = -\frac{\tau_w}{H}. \tag{4.14}$$

This equation describes the overall momentum balance between pressure gradient and skin friction.
By insertion of (4.14) into the right-handside of the equation (4.11) we obtain

$$\frac{\partial}{\partial y}\left(\langle u'v' \rangle - \nu\frac{\partial U}{\partial y}\right) = \frac{u_\tau^2}{H}. \tag{4.15}$$

In this equation H and u_τ are known and $U, \langle u'v' \rangle$ are unknown. Since there are more unknown quantities than equations, this problem is unclosed.

The logarithmic law

We will present a way of deriving the famous *log law*. In the previous paragraph we witnessed that equation (4.15) is unclosed since $U, \langle u'v' \rangle$ are unknown. In 1877 Boussinesq proposed the so-called *turbulent-viscosity hypothesis* [21, p.358] which states that the term $(-\langle u_i' u_j' \rangle + \frac{2}{3} k \delta_{ij})$ is proportional to the mean rate-of-strain

$$-\langle u_i' u_j' \rangle + \frac{2}{3} k \delta_{ij} = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) = 2 \nu_t \overline{S}_{ij},$$

where k denotes the turbulent kinetic energy which is defined to be half the trace of the Reynolds-stress tensor

$$k \equiv \frac{1}{2} \langle \vec{u}' \cdot \vec{u}' \rangle = \frac{1}{2} \sum_{i=1}^{3} \langle u_i' u_i' \rangle,$$

together with an algebraic model for the so-called *turbulent viscosity* or *eddy viscosity* ν_t designed for wall-bounded flows

$$\nu_t = \nu_t \left(U, \frac{\partial U}{\partial y}, y \right).$$

In the channel flow the mean-velocity is in the x-coordinate direction, while the solution only depends on the y-coordinate. Therefore the turbulent-viscosity hypothesis applied to $\langle u'v' \rangle$ is reduced to

$$\langle u'v' \rangle \;=\; -\nu_t \frac{\partial U}{\partial y}. \tag{4.16}$$

The turbulent viscosity ν_t is of dimension $m^2 s^{-1}$. This motivates the *ansatz* to write ν_t as a product of a characteristic velocity u_* and a characteristic length l_m

$$\nu_t = u_* l_m.$$

For the characteristic velocity u_* we will take the friction velocity u_τ. The natural length scale for a wall bounded flow is equal to the wall-distance y

$$u_* = u_\tau, \qquad l_m = \kappa y.$$

κ is a constant and will be determined later. From this we obtain for $\langle u'v' \rangle$

$$\langle u'v' \rangle = -\nu_t \frac{\partial U}{\partial y}, \qquad \nu_t = u_\tau \kappa y. \tag{4.17}$$

Due to the no-slip boundary condition we have $\vec{u} = 0,\ \vec{U} = 0,\ \vec{u}' = 0$ at the wall and hence $\langle u'v' \rangle = 0$ at the wall. If we reconsider equation (4.15) and integrate from $y' = 0$ to $y' = y$, we obtain

$$\begin{aligned} &\int_0^y \frac{\partial}{\partial y} \left(\langle u'v' \rangle - \nu \frac{\partial U}{\partial y} \right) dy' \\ &= -\int_0^y \frac{\partial}{\partial y} \left((\nu + \nu_t) \frac{\partial U}{\partial y} \right) dy' \\ &= -(\nu + \nu_t) \frac{\partial U}{\partial y}|_y + \nu \frac{\partial U}{\partial y}|_{y=0} = 0 \end{aligned}$$

as $\nu_t = u_\tau \kappa y$ vanishes at the wall $y = 0$. We obtain together with (4.13)

$$(\nu + \nu_t)\frac{\partial U}{\partial y}|_y = \nu \frac{\partial U}{\partial y}|_{y=0} = \frac{\tau_w}{\rho}.$$

We will now introduce the so-called universal coordinates (plus-units) obtained by using a coordinate transformation in y-coordinate direction. The equation in the bounded channel can also be written in these universal coordinates

$$u^+ = \frac{U}{u_\tau}, \quad y^+ = \frac{y u_\tau}{\nu}, \quad \nu_t^+ = \frac{\nu_t}{\nu}. \tag{4.18}$$

Note that the univeral coordinates are dimensionless.

If we use the plus-units then the chain rule for differentiation gives

$$\begin{aligned}
(\nu + \nu_t)\frac{\partial U}{\partial y} &= \frac{\tau_w}{\rho} \\
\Leftrightarrow (\nu + \nu_t)\frac{\partial U}{\partial u^+}\frac{\partial u^+}{\partial y^+}\frac{\partial y^+}{\partial y} &= \frac{\tau_w}{\rho} \\
\Leftrightarrow (\nu + \nu_t)\frac{u_\tau^2}{\nu}\frac{\partial u^+}{\partial y^+} &= \frac{\tau_w}{\rho} \\
\Leftrightarrow \frac{(\nu + \nu_t)}{\nu}\frac{\partial u^+}{\partial y^+} &= \frac{1}{u_\tau^2}\frac{\tau_w}{\rho} \\
\Leftrightarrow (1 + \nu_t^+)\frac{\partial u^+}{\partial y^+} &= 1.
\end{aligned} \tag{4.19}$$

In order to derive the log-law, we will make the following assumptions:

- We already mentioned that $\langle u'v' \rangle = 0$ at the wall. An additional consequence of (4.16) is that ν_t is much smaller than ν for $y^+ \lessapprox 10$ (close to the wall).
- In this section we assume that $\nu + \nu_t \approx \nu$ for $y^+ < 5$ and $\nu + \nu_t \approx \nu_t$ for $y^+ > 40$ and smooth in between.
- The non-dimensional wall-distance $y^+ = y \frac{u_\tau}{\nu}$ can be interpreted as a local Reynolds number. Therefore for small y^+-values (very close to the wall) we can assume laminar flow, whereas for large y^+-values fully developed turbulent flow conditions are expected.

Viscous sublayer

$y^+ \lessapprox 5$: Very close to the wall. As $\langle u'v' \rangle|_{y=0} = 0$ we have $\nu_t \ll \nu$ and thus from (4.18) and (4.19) we can deduce that:

$$\nu_t \ll 1 \Leftrightarrow \frac{\partial u^+}{\partial y^+} = 1 \Leftrightarrow u^+ = y^+.$$

Logarithmic boundary layer

$y^+ \gtrapprox 40$: At a sufficient large distance from the wall, $\nu_t \gg \nu$ and thus from (4.18) and (4.19) it follows:

$$\nu_t^+ = \kappa y^+ \gg 1 \Leftrightarrow \frac{\partial u^+}{\partial y^+} = \frac{1}{\kappa y^+} \Leftrightarrow u^+ = \frac{1}{\kappa}\ln y^+ + C. \tag{4.20}$$

The values of the constants κ and C are determined experimentally: $\kappa = 0.41$ and $C = 5.1$.
By means of these considerations we obtain the important *law of the wall*.

Theorem 4.2.1. *Law of the Wall*
In the region $y^+ \gtrapprox 40$ *and* $\frac{y}{H} \lessapprox 2$ *the universal velocity* u^+ *depends logarithmically on the universal coordinate* y^+*, i.e. the relation* $u^+ \sim ln\, y^+$ *holds. The equation* (4.20) *is called* logarithmic law of the wall *and the region where* (4.20) *holds is called* logarithmic part of the boundary layer.

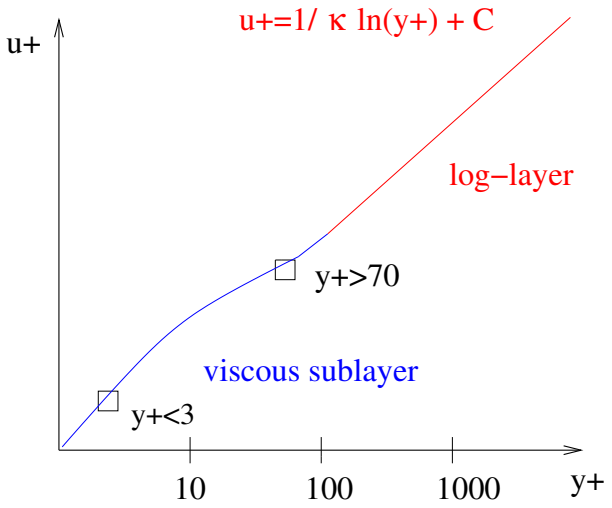

FIGURE 24: Schematic of the universal law of the wall.

Asymptotically the DNS data and the data from the schematic of the law of the wall are congruent with each other.

4.3 Test case setup for $Re_\tau = 395$

We will now present the test case setup for the channel flow. We will consider a Reynolds number $Re_\tau = 395$, based on the friction velocity and the channel halfwidth. For the flow DNS data by Moser, et al. [20] are available as reference data.

Computational grid
The computational domain is $(0, 2\pi)\times(0, 2)\times(0, \pi)$, thus we have $L = 2\pi$, $H = 1$ and $b = \pi$. For the computational grid a fine grid resolution near the wall is necessary, as near the wall, steep gradients exist, which need to be recovered sufficiently well. The mesh sizes in streamwise x- and spanwise z-direction are uniform. The grid in wall normal direction is stretched, using a hyperbolic tangent function

$$\frac{y(j)}{H} = \frac{\tanh[\gamma(2j/N_2 - 1)]}{\tanh(\gamma)} + 1.0, \qquad j = 0, \ldots, N_2 - 1, \tag{4.21}$$

where $y(j) \in [0.0, 2.0]$ is the coordinate of the jth grid point in y-coordinate direction. The stretching parameter γ is taken to be 2.3, 1.72, 1.5 and 1.2. The distance

of the shifted (first) off-wall point in wall-units is $y^+(1) = 0.79$. The $x-$, $y-$ and z-coordinate axes are discretized with $N \times N \times N$ nodes, where N is taken to be 16, 32, 48, 64. Figure 25 shows the computational grid with $N = 64$ and $\gamma = 2.2$.

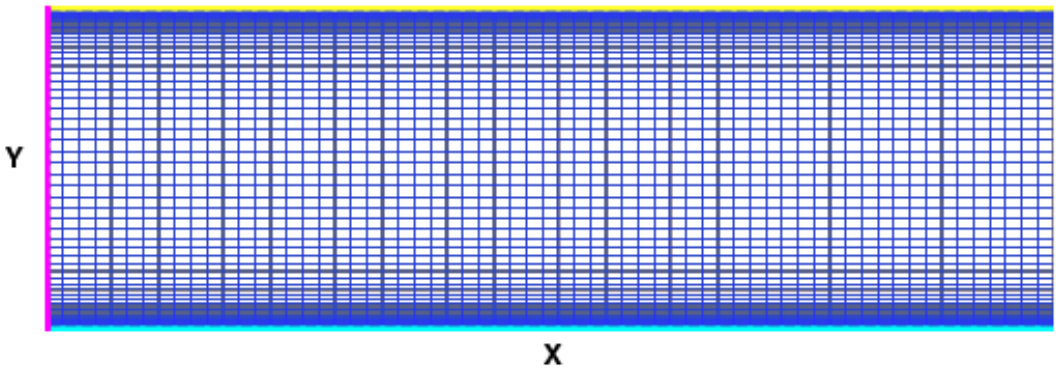

FIGURE 25: Mesh of a channel.

Computational time step
Chow and Moin propose in [2] a computational time step size $\Delta t^+ = 0.4$. We obtain with $\nu = 1.53 \cdot 10^{-5} \frac{m^2}{s}$ and $\rho = 1kg \cdot m^3$

$$\Delta t^+ = \Delta t \frac{u_\tau^2}{\nu},$$
$$u_\tau = Re_\tau \frac{\nu}{H} = 395 \ \frac{1.53 \cdot 10^{-5} \frac{m^2}{s}}{1m} = 6.0410^{-3} \frac{m}{s}.$$

Therefore Δt is chosen by

$$\Delta t = \delta t^+ \frac{\nu}{u_\tau^2} = 0.4 \frac{1.5310^{-5} \frac{m^2}{s}}{36.4810^{-6} m^2} \approx 0.4 \cdot 0.42 \approx 0.17s.$$

According to this we use a time step size $\Delta t = 0.17s$ in our computations.

Initial Conditions
The universal velocity profile $F(y^+)$ by Reichardt gives a good approximation to the DNS data by Moser et al [20]. The function $F(y^+)$ is given by

$$F(y^+) = \frac{1}{\kappa} ln(1 + 0.4y^+) + 7.8 \left[1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11} e^{-\frac{y^+}{3}} \right].$$

Here, Reichardt's function is adopted as the mean initial velocity,

$$U_R(y) = F(y^+) u_\tau.$$

A perturbation of 10 % of the bulk velocity U_{bulk} is added to $U_R(y)$ to form the initial velocity. The bulk velocity U_{bulk} is defined by:

$$U_{bulk} = \frac{1}{H} \int_0^H U dy.$$

Instead of U we take the velocity from Reichardt's function $U_R(y)$. We set the initial velocity $\vec{u}_0 = (u_0, v_0, w_0)$ to

$$\begin{aligned} u_0 &= U_R(y) + 0.1\ U_{bulk}\psi_{ran}, \\ v_0 &= 0.1\ U_{bulk}\psi_{ran}, \\ w_0 &= 0.1\ U_{bulk}\psi_{ran}, \end{aligned}$$

where ψ_{ran} is a random number in the interval $[-1, 1]$. Note that this velocity field is not divergence free. A divergence free velocity field is obtained after the first projection step.

Boundary Conditions
As we have already mentioned we have periodic boundary conditions in streamwise x-coordinate direction, no-slip boundary conditions at the walls in y-coordinate direction and symmetry boundary conditions in spanwise z-coordinate direction. Instead of prescribing $\vec{u} = \vec{0}$ at the viscous walls we impose

$$\vec{u} \cdot \vec{n} = 0, \quad (\mathbb{I} - \vec{n} \otimes \vec{n}) 2\nu \mathbb{S}(\vec{u})\vec{n} = -\tau_w \vec{u}_{t,\delta} \quad \text{on} \quad \Gamma_W.$$

We will explain the boundary conditions in detail in Section 4.4.

Pressure term p and driving force $\vec{f}$
The momentum equation reads

$$\frac{\partial \vec{u}}{\partial t} + \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) + \vec{\nabla} p - \vec{\nabla} \cdot (2(\nu + \nu_t)\mathbb{S}(\vec{u})) = \vec{f} \qquad \text{in} \qquad \Omega \times [0, T]\,.$$

The question is how to set pressure gradient $\vec{\nabla} p$ and source term $\vec{f}$. From (4.14) we have the relation between the *real* pressure gradient and wall shear stress in an infinitely long channel

$$\frac{\partial p_{real}}{\partial x} = \frac{\tau_w}{H}. \tag{4.22}$$

We already mentioned that the derivatives in y- and z-coordinate direction can be ignored. Therefore, we obtain for the real pressure gradient $\vec{\nabla} p_{real} = \frac{\partial p_{real}}{\partial x} = \frac{\tau_w}{H}\vec{e}_x$. In THETA we set $\vec{f} = \frac{\tau_w}{H}\vec{e}_x$ and $\vec{\nabla} p = 0$. So the driving force plays the role of the real pressure gradient. As we have periodic boundary conditions in streamwise direction the pressure gradient $\vec{\nabla} p$ may be computed using a difference quotient between both periodic layers, i.e.

$$\vec{\nabla} p \approx \frac{p_2 - p_1}{L}, \tag{4.23}$$

with p_1, p_2 being the values for p at $x_1 = 0$ and $x_2 = 2\pi$. As we have periodic boundary conditions the values at $x = 0$ and $x = 2\pi$ need to be the same, thus we have $p_1 = p_2$. With the computation of (4.23) we obtain $\vec{\nabla} p \approx 0$.
If the flow is only driven by the pressure gradient, the mass flux (i.e. the bulk velocity U_{bulk}) decreases unless $\vec{f}$ is chosen such that (4.22) is satisfied exactly. To keep the mass flux constant during a simulation, we have to adjust the driving force dynamically. Let $U_{bulk,THETA}(\delta t_n)$ be the bulk velocity of the computed solution of

the DLR-THETA-code at time step δt_n and $U_{bulk,DNS}$ be the desired bulk velocity obtained from the DNS data. The source term $\vec{f}$ is modified as follows

$$\vec{f}(\delta t_{n+1}) = \begin{pmatrix} \frac{u_\tau^2}{H} \\ 0 \\ 0 \end{pmatrix} + \frac{1}{\delta t_n} \begin{pmatrix} U_{bulk,DNS} - U_{bulk,THETA}(\delta t_n) \\ 0 \\ 0 \end{pmatrix}.$$

The correction term enforces the mass flux conservation. When $U_{bulk,THETA}(\delta t_n)$ is larger than $U_{bulk,DNS}$, the source term will decrease to make the flow slow down, otherwise it would speed up.

4.4 Solution of the Navier-Stokes equations at the boundary

In this section we will consider how to solve the Navier-Stokes equations at control volumes K_{ij} which lie at the boundary of the domain Ω (see also [12]). Note that $\vec{u} = \vec{0}$ at $K_{ij} \cap \partial\Omega$ but $\vec{u} \neq \vec{0}$ at K_{ij}. After application of the filtering which was introduced in Subsection 4.1.1 we are given the Navier-Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{\nabla} \cdot (\vec{u} \otimes \vec{u}) + \vec{\nabla} p - \vec{\nabla} \cdot 2(\nu + \nu_t)\mathbb{S}(\vec{u}) = \vec{f} \qquad \text{in} \qquad \Omega \times [0,T], \tag{4.24}$$

$$\vec{u} = 0 \qquad \text{in} \qquad \partial\Omega \times [0,T]. \tag{4.25}$$

Applying the finite volume method and the Gauß' theorem to the momentum equation, we obtain

$$\int_{K_{ij}} \frac{\partial \vec{u}}{\partial t} dV + \int_{\partial K_{ij}} (\vec{u} \otimes \vec{u}) \cdot \vec{n} dS + \int_{\partial K_{ij}} p \cdot \vec{n} dS - \int_{\partial K_{ij}} 2(\nu + \nu_t)\mathbb{S}(\vec{u})\vec{n} dS = \int_{K_{ij}} \vec{f} dV$$

for a control volume K_{ij}. Let Γ_W be the boundary of the domain Ω. For a control volume K_{ij} which lies adjacent to Γ_W we have

$$\overline{K_{ij}} \cap \Gamma_W \neq \varnothing.$$

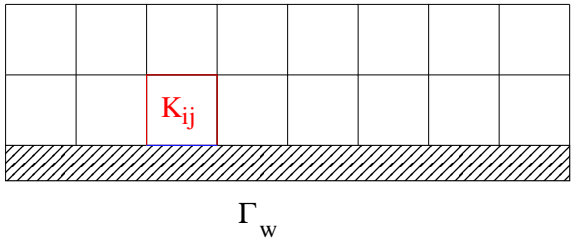

FIGURE 26: Schematic of control volume K_{ij} and wall Γ_w.

The momentum equation on $\overline{K}_{ij} \cap \Gamma_W$ reads

$$\begin{aligned} &\int_{\overline{K}_{ij} \cap \Gamma_W} \frac{\partial \vec{u}}{\partial t} dV + \int_{\partial K_{ij} \cap \Gamma_W} (\vec{u} \otimes \vec{u}) \cdot \vec{n} dS \\ &+ \int_{\partial K_{ij} \cap \Gamma_W} p \cdot \vec{n} dS - \int_{\partial K_{ij} \cap \Gamma_W} 2(\nu + \nu_t)\mathbb{S}(\vec{u})\vec{n} dS \qquad = \int_{\overline{K}_{ij} \cap \Gamma_W} \vec{f} dV. \end{aligned}$$

Considering each term we obtain

$$\int_{\overline{K}_{ij}\cap\Gamma_W} \frac{\partial \vec{u}}{\partial t} dV = 0 \quad \text{as } \vec{u} = 0 \text{ on } \Gamma_W,$$

$$\int_{\partial K_{ij}\cap\Gamma_W} (\vec{u}\otimes\vec{u})\cdot\vec{n} dS = 0 \quad \text{as } \vec{u} = 0 \text{ on } \Gamma_W,$$

$$\int_{\partial K_{ij}\cap\Gamma_W} 2(\nu+\nu_t)\mathbb{S}(\vec{u})\vec{n} dS \neq 0 \quad \text{as the symmetric stress tensor } \mathbb{S}(\vec{u}) \neq 0 \text{ on } \Gamma_W,$$

$$\int_{\partial K_{ij}\cap\Gamma_W} p\cdot\vec{n} dS \neq 0 \quad \text{as in the general case } p \neq 0 \text{ on } \Gamma_W.$$

From the turbulent-viscosity hypothesis (4.16) we obtain for ν_t

$$\nu_t \frac{\partial U}{\partial y}|_{y=0} \;=\; \langle u'v' \rangle \;=\; 0.$$

It follows that $\nu_t = 0$ as $\frac{du}{dy}|_{y=0} \neq 0$. Therefore the momentum equation reduces to

$$\int_{\partial K_{ij}\cap\Gamma_W} p\cdot\vec{n} dS - \int_{\partial K_{ij}\cap\Gamma_W} 2\nu\mathbb{S}(\vec{u})\vec{n} dS = \int_{K_{ij}\cap\Gamma_W} \vec{f} dV. \tag{4.26}$$

We will divide the vector $\mathbb{S}(\vec{u})\cdot\vec{n}$ into two components: the wall normal component and the wall-tangential component. Therefore, we need an operator which projects $\mathbb{S}(\vec{u})\cdot\vec{n}$ to a vector in wall-tangential and to a vector in wall normal direction. Let $\vec{x}$ be a vector $\in \mathbb{R}^3$. Let $\vec{n}$ be the normal vector in wall normal direction. Then $\vec{x}\cdot\vec{n}$ is a scalar containing the component of $\vec{x}$ in wall normal direction. For $(\vec{x}\cdot\vec{n})\cdot\vec{n}$ we obtain a vector in wall normal direction of the same magnitude as the wall normal component of $\vec{x}$.

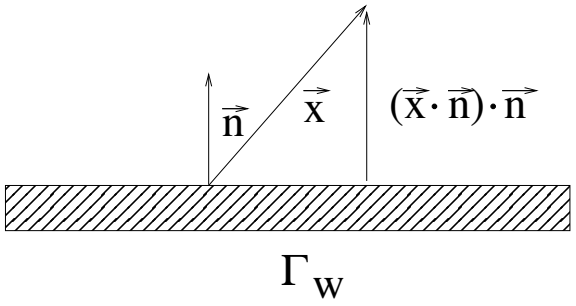

FIGURE 27: Projection of $\vec{x}$ to the wall normal direction.

Consequently if $\vec{n}$ is the normal vector in wall normal direction, then the vector

$$\vec{x} - (\vec{x}\cdot\vec{n})\vec{n} = \mathbb{I}\vec{x} - (\vec{n}\otimes\vec{n})\vec{x} = (\mathbb{I} - \vec{n}\otimes\vec{n})\vec{x}$$

contains the components of $\vec{x}$ parallel to the wall. If we consider again the symmetric rate-of-strain tensor in (4.26) we obtain

$$\int_{\partial K_{ij}\cap\Gamma_W} 2\nu\mathbb{S}(\vec{u})\cdot\vec{n} dS \tag{4.27}$$

$$= \underbrace{\int_{\partial K_{ij}\cap\Gamma_W} 2\nu(\vec{n}\otimes\vec{n})\mathbb{S}(\vec{u})\cdot\vec{n} dS}_{\text{component in wall normal direction},\approx 0} + \underbrace{\int_{\partial K_{ij}\cap\Gamma_W} 2\nu(\mathbb{I}-\vec{n}\otimes\vec{n})\mathbb{S}(\vec{u})\cdot\vec{n} dS}_{\text{component parallel to the wall}}. \tag{4.28}$$

In the following we will denote Γ_δ as an artificial inner boundary containing the shifted nodes at wall distance y_δ.

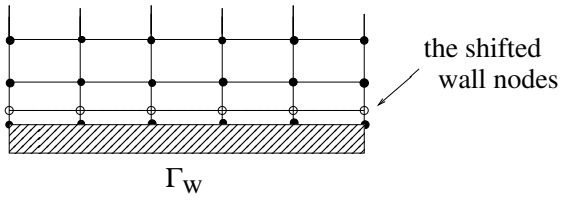

FIGURE 28: The shifted grid nodes near the boundary part.

Subsequently we define $\vec{v}_{t,\delta}$ as the projection of a vector $\vec{u}$ onto Γ_δ, $\vec{u}_{t,\delta}$ the normalization of it and u_δ its length:

$$\begin{aligned} \vec{v}_{t,\delta} &= (\mathbb{I} - \vec{n} \otimes \vec{n})\vec{u}|_{\Gamma_\delta} \\ \vec{u}_{t,\delta} &= \frac{\vec{v}_{t,\delta}}{||\vec{v}_{t,\delta}||} \\ u_\delta &= |\vec{v}_{t,\delta}| \end{aligned}$$

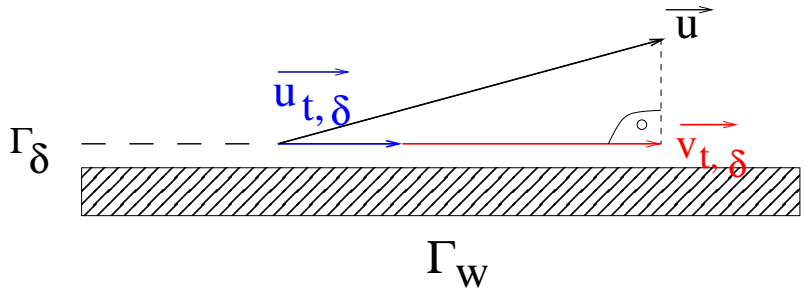

FIGURE 29: Artificial inner boundary Γ_δ and vectors $\vec{v}_{t,\delta}$ (red), $\vec{u}_{t,\delta}$ (blue) of velocity vector $\vec{u}$.

In the following we will assume a one-dimensional velocity field which only depends on y, i.e. $\vec{u} = (U(y), 0, 0)$. By means of this assumption we can modify the integrand of (4.27) with wall shear stress τ_w into:

$$\begin{aligned} & 2\nu(\mathbb{I} - \vec{n} \otimes \vec{n})\mathbb{S}(\vec{u}) \cdot \vec{n} \\ =& 2\nu(\mathbb{I} - \vec{n} \otimes \vec{n}) \begin{pmatrix} 0 & \frac{1}{2}\frac{\partial U}{\partial y} & 0 \\ \frac{1}{2}\frac{\partial U}{\partial y} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ =& \nu \begin{pmatrix} \frac{\partial U}{\partial y} \\ 0 \\ 0 \end{pmatrix} = \tau_w \vec{u}_{t,\delta}. \end{aligned}$$

For the pressure term we can compute the integral as:

$$\int_{\partial K_{ij} \cap \Gamma_W} p\vec{n} dS \approx p_{i,j}\ \vec{n} \cdot V(\partial K_{ij} \cap \Gamma_W),$$

where $V(\partial K_{ij} \cap \Gamma_W)$ denotes the surface area of the face $\partial K_{ij} \cap \Gamma_W$ and $p_{i,j}$ the value of pressure p at the cell K_{ij}. Thus we obtain together with an approximation of the right handside for the momentum equation at the wall

$$\int_{\partial K_{ij} \cap \Gamma_W} -\tau_w \vec{u}_{t,\delta} dS + p_{i,j} \vec{n} \cdot V(\partial K_{ij} \cap \Gamma_W) = \vec{f}_{i,j} \cdot V(K_{ij} \cap \Gamma_W).$$

4.5 Computation of the wall shear stress τ_w

The wall-shear stress τ_w is defined as

$$\tau_w = \nu \vec{\nabla} u_\delta \cdot \vec{n}|_{\Gamma_W} = \nu \frac{\partial u_\delta}{dy}|_{y=0}.$$

There are two possible methods to compute the derivative:

1. Let y_δ^+ denotes the +-coordinate of the shifted node. Applying the law of the wall it follows that if $y_\delta^+ \lesssim 2$, then we have $y_\delta^+ = u_\delta^+$, i.e. we have a linear relation between u_δ^+ and y_δ^+. Thus the approximation of $\frac{\partial u_\delta}{\partial y}|_{y=0}$, using the difference quotient

$$\frac{\partial u_\delta}{\partial y}|_{y=0} \approx \frac{u_\delta - u|_{y=0}}{y_\delta - y|_{y=0}} = \frac{u_\delta}{y_\delta} \tag{4.29}$$

   gives reliable results.

2. If we have $y_\delta^+ \gtrapprox 2$ the approximation with the difference quotient is bad, as we do not have a linear relation between u_δ^+ and y_δ^+. In this case τ_w is computed from friction velocity u_τ. Assuming that there is a relation between u_δ^+ and y_δ^+, such that there exists a function F with

$$F(y_\delta^+) = F(\frac{y_\delta u_\tau}{\nu}) = u_\delta^+ = \frac{u_\delta}{u_\tau} \ .$$

   To compute the friction velocity u_τ we will consider the function

$$g(u_\tau) = u_\tau F(\frac{y_\delta u_\tau}{\nu}) - u_\delta = u_\tau F(y_\delta^+) - u_\delta \tag{4.30}$$

   and search the zero(s) of this function, i.e. search u_τ such that $g(u_\tau) = 0$. A proper method to find the zero(s) is Newton's method:

   If we seek x such that $g(x) = 0$, we will consider the iterative scheme

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \ . \tag{4.31}$$

   Then $x_n \to x$ as $n \to \infty$ with $g(x) = 0$.

   From the definition in (4.30) and the chain rule and product rule for differentiation we obtain

$$g'(u_\tau) = F(\frac{y_\delta u_\tau}{\nu}) + u_\tau F'(\frac{y_\delta u_\tau}{\nu}) \frac{y_\delta}{\nu} = F(y_\delta^+) + F'(y_\delta^+) y_\delta^+ \ . \tag{4.32}$$

Using the method of successive approximations (4.31) we obtain the zero of the function g, i.e. u_τ. Since $g(u_\tau)$ is a monotonically increasing function, its zero is uniquely determined. From velocity u_τ we then compute τ_w with

$$\tau_w = \rho \cdot u_\tau^2 \quad \text{as} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}} \ . \tag{4.33}$$

How to choose a proper function F? An approach would be to derive it from a spline function (B-spline) to approximate the DNS data of y^+ and u^+. An alternative approach is to use the universal velocity profile $F(y^+)$ by Reichardt given by:

$$F(y^+) = \frac{ln(1+0.4y^+)}{\kappa} + 7.8(1 - e^{-\frac{y^+}{11.0}} - {}^{y^+}/_{11.0e^{-\frac{y^+}{3.0}}}),\ \kappa = 0.4. \tag{4.34}$$

In the implementation of THETA a single formula for an auxiliary eddy-viscosity ν_{rd} at the shifted node y_δ is used to compute τ_w:

$$\tau_w = \nu_{rd}\frac{u_\delta}{y_\delta} \quad \text{with} \quad \nu_{rd} = \nu \ \max\left(1, \frac{y_\delta^+}{F(y_\delta^+)}\right) \ . \tag{4.35}$$

We will denote $y^+ \equiv y_\delta^+$ for simplicity. The formula (4.35) and the if/else switch concerning the plus-unit of the first shifted wall node give the same results. The if-else switch reads as follows:

If $y^+ < 2$ compute τ_w from (4.29), else compute τ_w from Reichardt function including the steps (4.30), (4.31), (4.32), (4.33).
In THETA, this if-else statement is formulated as follows:
If $\max(1, \frac{y^+}{F(y^+)}) = 1$, compute τ_w from

$$\tau_w = \nu_{rd} \cdot \frac{u_\delta}{y_\delta} = \nu\frac{u_\delta}{y_\delta} \ .$$

This is equivalent to the relation in (4.29) using the difference quotient.

If $\max\left(1, \frac{y^+}{F(y^+)}\right) = \frac{y^+}{F(y^+)}$, compute τ_w from

$$\tau_w = \nu_{rd} \cdot \frac{y^+}{F(y^+)}\frac{u_\delta}{y_\delta} = \nu\frac{\frac{y_\delta u_\tau}{\nu}}{F(y^+)} \cdot \frac{u_\delta}{y_\delta} = \frac{u_\tau u_\delta}{F(y^+)}.$$

With the definition of $F(y^+) = \frac{u_\delta}{u_\tau}$ we obtain

$$\tau_w = \frac{u_\tau u_\delta}{F(y^+)} = u_\tau^2.$$

This is in agreement with (4.33) since we assumed that $\rho \equiv 1$. We have seen that the theoretical approach which switches between $y^+ > 2$ and $y^+ < 2$ leads to the same results like the implementation used in THETA.

Now we will give a brief explanation for the equivalence of both case separations. The following is to show:

$$y^+ \lesssim 2 \longrightarrow \max\left(1, \frac{y^+}{F(y^+)}\right) = 1 \ ,$$
$$y^+ \gtrapprox 2 \longrightarrow \max\left(1, \frac{y^+}{F(y^+)}\right) = \frac{y^+}{F(y^+)} \ .$$

The answer is given by the law of the wall :
If $y^+ \lesssim 2$ it follows that $F(y^+) = u^+ \approx y^+$ and consequently $\frac{y^+}{F(y^+)} = 1 = max(1, \frac{y^+}{F(y^+)})$. If $y^+ \gtrsim 2$ it follows that $F(y^+) = u^+ \sim ln(y^+)$ and so $u^+ < y^+$. Consequently, $\frac{y^+}{F(y^+)} > 1$, and therefore, $\frac{y^+}{F(y^+)} = max(1, \frac{y^+}{F(y^+)})$.

N.B. 4.5.1. *Note that the coordinate of the shifted node is important for the way of computation of τ_w. If $y_\delta^+ \gtrsim 2$ we have to use the function of Reichardt in order to compute the wall shear stress τ_w. Else if $y_\delta^+ \lesssim 2$ use the linear difference quotient. How can the condition be translated onto the mesh? The y^+-coordinate of the first shifted node is computed from the y^+-coordinate of the first node of the grid $y^+(1)$*

$$y_\delta^+ = 0.27 \cdot y^+(1).$$

The plus-unit is computed by

$$y^+ = \frac{u_\tau}{\nu} y.$$

From the condition for the shifted node we obtain for the coordinate of the first node, setting $\frac{u_\tau}{\nu} = 395$

$$\begin{aligned} y_\delta^+ &\lesssim 2 \\ \Leftrightarrow 0.27 y^+(1) &\lesssim 2 \\ \Leftrightarrow 0.27 \cdot \frac{u_\tau}{\nu} \cdot y(1) &\lesssim 2 \\ \Leftrightarrow y(1) &\lesssim 0.018. \end{aligned} \tag{4.36}$$

As a result we can determine from the y-coordinate of the first node whether or not to use a so-called wall function.

4.6 Statistical turbulence modelling

The solution of turbulent flows is unsteady and appears random-like. Therefore, for postprocessing of a large-eddy simulation a suitable averaging of the velocity field is needed. Averaging the solution of the velocity field is done by applying an averaging filter to the velocity field $\vec{u}$

$$\langle \cdot \rangle : \vec{u} \longrightarrow \langle \vec{u} \rangle.$$

In the following we will present different ways of filtering.

Definition 4.6.1. *The time averaging filter $\langle . \rangle_{[t,t+\Delta t]}$ over the time interval $[t, t+\Delta t]$ is defined by*

$$\langle \vec{u} \rangle_t = \langle \vec{u} \rangle_{[t,t+\Delta t]} = \frac{1}{\Delta t} \int_t^{t+\Delta t} \vec{u}(x, y, z, t) dt.$$

Let $\vec{u}(x, y, z, t_n)$ be the values of the velocity field at N timesteps $(t_1, \ldots, t_N)$ with $t_n = n\Delta t$ for $n = 1, \ldots, N$. Then the discrete time averaging filter $\langle . \rangle_{t_N}$ over the interval $[t_1, t_N]$ is defined by

$$\langle \vec{u} \rangle_{t_N} = \frac{1}{N} \sum_{n=1}^{N} \vec{u}(x, y, z, t_n).$$

As a result, the filtered velocity field $\langle\vec{u}\rangle_{t_N}$ becomes independent of time, i.e.

$$\langle\vec{u}\rangle_{t_N} \equiv \langle\vec{u}(x,y,z,t)\rangle_{t_N} = \langle\vec{u}(x,y,z)\rangle_{t_N}.$$

The second way of filtering happens not concerning time but space, in particular concerning the homogenous directions of a flow field. The following definition holds for the channel flow which has two homogenous directions, in x- and z-coordinate direction.

Definition 4.6.2. *The space averaging filter $\langle.\rangle_{[x,x+\Delta x]\times[z,z+\Delta z]}$ in the homogenous streamwise x-coordinate direction and the spanwise z-coordinate direction over the interval $[x, x+\Delta x]\times[z, z+\Delta z]$ is given by*

$$\langle\vec{u}\rangle_s = \langle\vec{u}\rangle_{[x,x+\Delta x]\times[z,z+\Delta z]} = \frac{1}{\Delta x \Delta z}\int_x^{x+\Delta x}\int_z^{z+\Delta z}\vec{u}(x,y,z,t)dzdx.$$

Let $\vec{u}(x_i, y, z_j, t)$ for $i = 1,\ldots,N_x$, $j = 1,\ldots,N_z$ be the values of the velocity field at N_x discrete points in x-coordinate direction ($x_i = i\Delta x$) and N_z discrete points in z-coordinate direction ($z_j = j\Delta z$). Then the discrete space averaging filter $\langle.\rangle_{(N_x,N_z)}$ over the interval $[x_1, x_{N_x}]\times[z_1, z_{N_z}]$ is defined by

$$\langle\vec{u}\rangle_{(N_x,N_z)} = \frac{1}{N_x \cdot N_z}\sum_{i=1}^{N_x}\sum_{j=1}^{N_z}\vec{u}(x_i, y, z_j, t).$$

In Figure 30 the principle of spatial averaging in x- and z-coordinate direction is sketched.

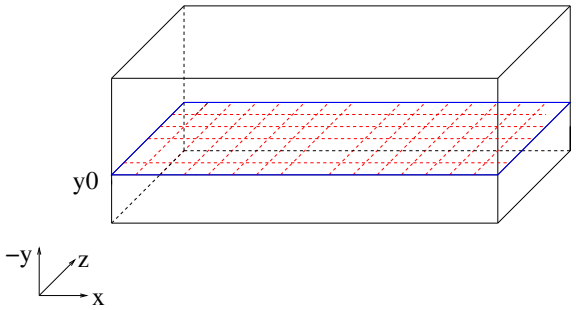

FIGURE 30: Computation of the average value over all points in the xz-plane where $y = y_0$ (marked).

As a result we obtain for each fixed y an average value over the xz-plane. Consequently the averaged velocity field $\vec{u}$ does *not* depend on the values in x- and z-direction

$$\langle\vec{u}\rangle_{(N_x,N_z)} \equiv \langle\vec{u}(x,y,z,t)\rangle_{(N_x,N_z)} = \langle\vec{u}(y,t)\rangle_{(N_x,N_z)}.$$

Our aim is to combine the spatial and time averaging and therefore to obtain the mean velocity field $\langle\vec{u}(x,y,z,t)\rangle = \langle u(y)\rangle = U(y)$ which only depends on y.

Definition 4.6.3. *A turbulent channel flow is called fully developed and statistically steady (statistically converged) if the following limit values exist*

$$\lim_{\Delta t\to\infty}\langle\vec{u}\rangle_{[t,t+\Delta t)}, \qquad \lim_{\Delta z\to\infty}\langle\vec{u}\rangle_{[x,x+\Delta x]\times[z,z+\Delta z)}$$

where in the following we keep Δx constant and sufficiently large.

After a certain time the solution of the velocity field in the channel becomes statistically steady. The time required to achieve a statistically steady state is equal to a certain streamwise distance from the channel entrance. The following Ergodic theorem [1] states that averaging over time and over the homogeneous coordinate directions gives the same solution, which is therefore two-dimensional and steady-state.

Theorem 4.6.4. *(Ergodic theorem)*
Consider a fully developed turbulent channel flow, which is homogeneous in streamwise x- and spanwise z-coordinate direction. Then

$$\lim_{\Delta t\to\infty}\langle\vec{u}\rangle_{[t,t+\Delta t)} = \lim_{\Delta z\to\infty}\langle\vec{u}\rangle_{[x,x+\Delta x]\times[z,z+\Delta z)}.$$

As time-averaging and space-averaging are the same we may combine space and time averaging and define the statistical averaging operator $\langle . \rangle$ by:

$$U(y) \equiv \langle\vec{u}\rangle = \langle\langle\vec{u}\rangle_t\rangle_s = \langle\langle\vec{u}\rangle_{[t+\Delta t]}\rangle_{[x,x+\Delta x]\times[z,z+\Delta z]}.$$

Our objective is to run the test case channel flow using the DLR-THETA-code and to apply the filter $\langle\langle\vec{u}\rangle_t\rangle_s$.
In the simulation we let the flow develop until we reach a statistically steady state at $t = T_{start}$. We now have to apply time and space filter operations. During the THETA run, $\langle\vec{u}\rangle_t$ is computed as a running average. After the simulation a postprocessing tool is used to compute the average over the spanwise directions, $\langle\langle\vec{u}\rangle_t\rangle_s$. We will use the following formula for averaging

$$\langle\vec{u}\rangle \approx \langle\langle\vec{u}\rangle_s\rangle_t = \frac{1}{N_x N_z (T_{stop}-T_{start})}\sum_{k=1}^{N_z}\sum_{i=1}^{N_x}\left(\sum_{l=T_{start}}^{T_{stop}}\vec{u}(x_i,y,z_k,t_l)\right).$$

Note that the averaging filter has the same properties as the scale separation filter $\mathcal{T}$ in Subsection 4.1.1.

4.7 Interpretation of the fluctuations

With the filter we can decompose the velocity field $\vec{u}(\vec{x})$ into a mean part $\langle\vec{u}(y)\rangle$, which is time independent and, in the case of turbulent channel flow, does not depend on x- and z-coordinates and into a fluctuation velocity field denoted by $\vec{u}'$, i.e.

$$\vec{u}(\vec{x}) = \langle\vec{u}(y)\rangle + \vec{u}'(\vec{x},t).$$

The average value for the fluctuation part is zero, i.e.

$$\langle\vec{u}'\rangle \;=\; \vec{0}.$$

The fluctuations are time-dependent as they denote the difference between statistical averaged solution $\langle\vec{u}(y)\rangle$ and the time dependent solution at each time step $\vec{u}(\vec{x},t)$

$$\vec{u}'(\vec{x},t) = \vec{u}(\vec{x},t) - \langle\vec{u}(y)\rangle.$$

In THETA, the average quadratic fluctuations $\langle\vec{u}_i'\vec{u}_j'\rangle$ $(i,j = 1,2,3)$ are computed at each mesh point over time interval $[0,T]$. Spatial averaging in both homogenous directions is computed during postprocessing. Consequently, we obtain the average (in sense of time and space) quadratic fluctuations depending on the inhomogenous coordinate direction y. The statistical *variance* is defined as:

$$Var(X) = E(X - E(X))^2$$

where X denotes a random variable and $E(X)$ its expected value (or mean value). The average quadratic fluctuations $\langle\vec{u}_i'\vec{u}_j'\rangle$ can be interpreted as the variance of the flow field velocity. The standard deviation σ_X denotes the square root of the variance for a random variable X

$$\sigma_X := \sqrt{Var(X)}.$$

A normally distributed random variable is completely described by its expected value $E(X)$ and its standard deviation σ_X. Assuming that the velocity $\vec{u}$ is normally distributed, from the standard deviation we obtain thus:

- 68 % of the velocity has the value in the interval $[E(X) - \sigma_X, E(X) + \sigma_X]$,
- 95.4 % of the velocity has the value in the interval $[E(X) - 2\sigma_X, E(X) + 2\sigma_X]$,
- 99.7 % of the velocity has the value in the interval $[E(X) - 3\sigma_X, E(X) + 3\sigma_X]$.

Using the variance it is possible to predict the spectrum of the values of the velocity. For the channel flow we have a reference velocity of $0.12\frac{m}{s}$. For a large variance we may say that larger fluctuations of the velocity are possible, whereas for a small variance the velocity does not differ much from the expected value $E(X)$. A big variance also indicates regions where the flow is unsteady and turbulent.

5. Validation of Kármán-sensor for turbulent channel flow

We intend to develop a sensor which allows a CFD code user to assess the quality of the grid for an LES simulation. The sensor should be able to indicate if a grid is too coarse or if it is fine enough for a computation. Since the sensor can be evaluated using the LES solution on one grid, it is sometimes called single-grid estimator. From the viewpoint of application of LES for large-scale problems in research and industry, this is the typical situation, since simulations on successively refined meshes can often not be afforded.

In LES, an approximation to the turbulent solution of the Navier-Stokes equations is computed. This solution is unsteady and three-dimensional so that the numerical method has to resolve the gradient of the solution in all three coordinate directions. Therefore, the grid spacings Δx, Δy, and Δz are important in an LES.

In contrast, a RANS simulation only gives an approximation to the time-averaged solution of the Navier-Stokes equations. For turbulent channel flow, the RANS solution can be described using Reichardts law (4.34), which only depends on the wall normal coordinate y. Therefore, the quality of a RANS simulation e.g. for channel flow is determined only by the numerical solution in wall normal direction y. In the following sections we will try to answer the following questions:

1. Which parameter of the grid dominates the quality of an LES simulation?
2. How can the Kármán-sensor be used to estimate the quality of a mesh?

In the following sections we will present the numerical results of the channel flow obtained by an LES simulation with the DLR-THETA-code. The quality will be assessed by comparing it to the data of a direct numerical simulation. We will deduce which parameter of the grid is important for reliable numerical results and how computations could be more efficient. Thereafter we will show how the Kármán-sensor is computed for channel flow and present the numerical results. The sensor will be compared to several grid parameters and from this we will try to derive a first hypothesis in order to answer the second question.

5.1 Precursor study on statistical averaging and grid resolution

5.1.1 Convergence behavior for time averaging

In this section we will consider the question at which point in time the statistical averaging should start and how long the time interval for statistical averaging should be. In the previous chapter we already mentioned that it takes some time for the flow development of the channel to become statistically steady. For this reason the statistical averaging is not started from $t = 0$. To determine an optimal interval for statistical averaging we perform several computations on relatively fine meshes with 64^3 and 48^3 mesh points, averaging from t_0 until t_n where t_n denotes the ending time step of the computation. Our time step size is $\Delta t = 0.17s$ thus the time interval size of each computation is $T = t_n \cdot \Delta t$. We perform the following computations:

- Averaging from $t_0 = 1000$ until $t_n = 3000$.
- Averaging from $t_0 = 4000$ until $t_n = 8000$.
- Averaging from $t_0 = 8000$ until $t_n = 16000$.
- Averaging from $t_0 = 8000$ until $t_n = 24000$.
- Averaging from $t_0 = 8000$ until $t_n = 32000$.

We will plot the universal coordinates y^+ and U^+ and fluctuation $\langle u'u' \rangle^+$. For reference data we will take the DNS data of Moser, et al.[20]. The numerical results are shown in the following graphs.

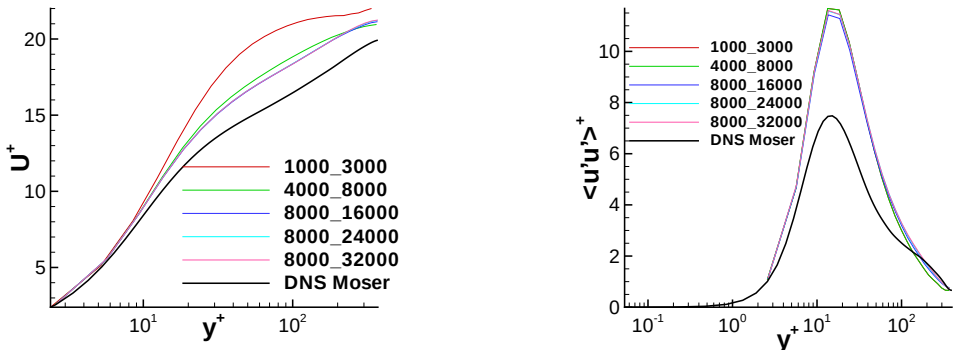

FIGURE 31: y^+ and U^+ for N=48 (left), fluctuations $\langle u'u' \rangle^+$ (right) for various averaging intervals.

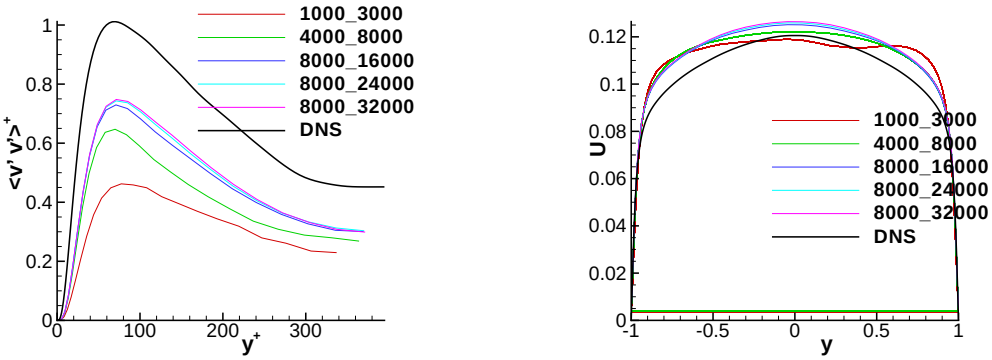

FIGURE 32: Fluctuations $\langle v'v' \rangle^+$ (left) and mean velocity U (right) for various averaging intervals.

The statistical convergence of the time-averaged LES data can be clearly seen from Figures 5.1 and 5.2. The solutions averaged over the time interval $[t_0\Delta t, t_n\Delta t]$ show convergence for t_0 sufficiently large, say $t_0 = 8000$, and $t_n \to \infty$. Since $t_n \to \infty$ cannot be realized, we use the fact that the sequence of solutions averaged over $[t_0\Delta t, t_n\Delta t]$ clearly forms a Cauchy sequence. The predictions for $\langle v'v' \rangle$ require a larger t_n: The changes between the solution averaged over 8000-32000 time steps and over 8000-24000 time steps are already very close. However, regarding the mean velocity profile, U, and the streamwise fluctuations, $\langle u'u' \rangle$, reasonable convergence has been achieved for averaging over the 8000-16000 time steps.

How can we imagine the meaning of 8000 time steps?
One time step is 0.17 s. From the DNS data we obtain as reference velocity $U_{ref} \approx 0.12\frac{m}{s}$. The distance L the fluid has passed until the statistical averaging at time $t_0 \cdot \Delta t = 1360s$ begins is therefore approximately $231m$. Since the length, l, of the channel is $6.28m$, we observe that during 1360 s the flow has traveled 36 times through the channel. From this we may say that 8000 time steps are equivilant to $1360s \approx 20$ minutes, or 36 flow-through times.

5.1.2 Convergence behavior for isotropic grid refinement

The second issue we will consider for this precursor study is the mesh size. First we will consider isotropic grid refinement, i.e., the refinement in all three coordinate directions is almost identical. The time step size is $\Delta t = 0.17s$ and statistical averaging is computed in the time interval between $t_0 = 8000$ and $t_n = 16000$. The results are shown in the following figures.

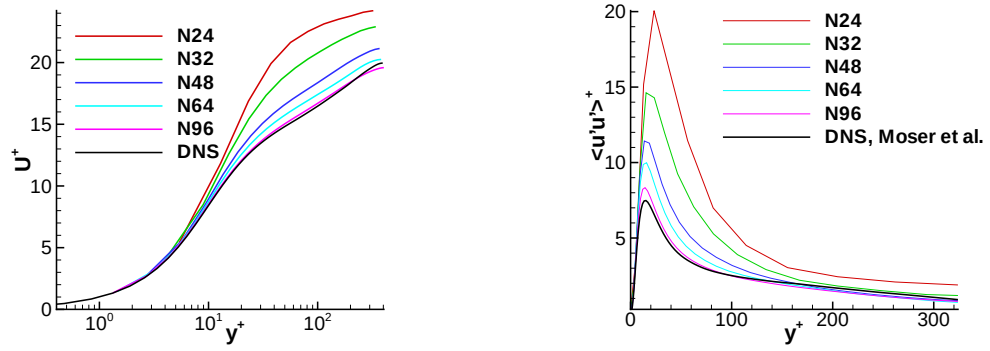

FIGURE 33: y^+ and U^+ (left), fluctuations $\langle u'u' \rangle^+$ (right) with statistical averaging between t_0=8000 and t_n=16000 for various mesh widths.

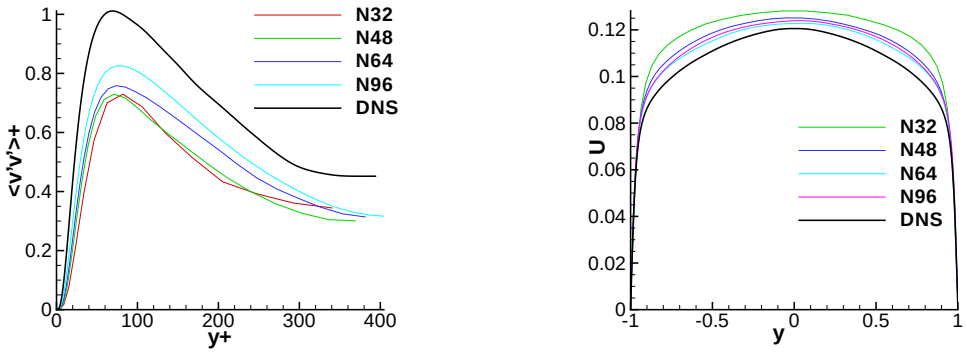

FIGURE 34: Fluctuations $\langle v'v' \rangle^+$ (left) and mean velocity U (right) for various mesh widths.

The results for the meshes with $N = 24$ and $N = 32$ (Fig. 5.3 and 5.4) differ much from the experimental solution of the DNS data. The results for a mesh with $N = 48$ displays harmony with the experimental solution, but the solution for the finer meshes with $N = 64$ and $N = 96$ is better yet. As a consequence we can say that a computation on a mesh with $N = 48$ supplies acceptable results. Computations on coarser grids are not recommended.

5.1.3 Convergence behavior for non-isotropic grid refinement

In this section we will investigate whether a grid refinement in one or two directions (= non-isotropic grid refinement) can improve the computation and which coordinate direction has the largest influence on the results. From this we want to obtain an answer to the first question of this chapter, i.e. which grid parameter is dominating the quality of an LES simulation. Therefore, we generated several grids with $N_x \times N_y \times N_z$ nodes where $N_x, N_y, N_z \in \{48, 64, 96\}$. We run the computation of the channel flow with statistical averaging between $t_0 = 8000$ and $t_n = 16000$ and compare the results of the mean velocity U to the DNS data. The results are displayed in the following graphs.

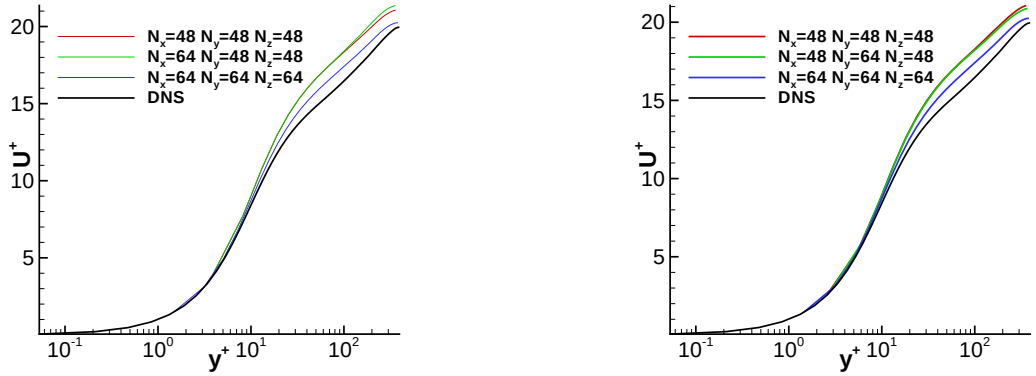

FIGURE 35: y^+ and U^+ for meshes with grid refinement in x- (left) and y-coordinate direction (right).

First we will consider refinement in either x- or y-coordinate direction (Fig. 5.5). As reference solution, we will take the solution on the 48^3 mesh and the 64^3 mesh. A mesh refinement in x-coordinate direction alone does not improve the results, if Δy and Δz are kept constant (see Fig. 5.5 (left)). Similarly, a refinement only in y-coordinate direction does not improve the results (see Fig. 5.5 (right)). As the costs of a computation increase with refinement in one direction it is not advisable to refine the grid in x- or y-coordinate direction alone.

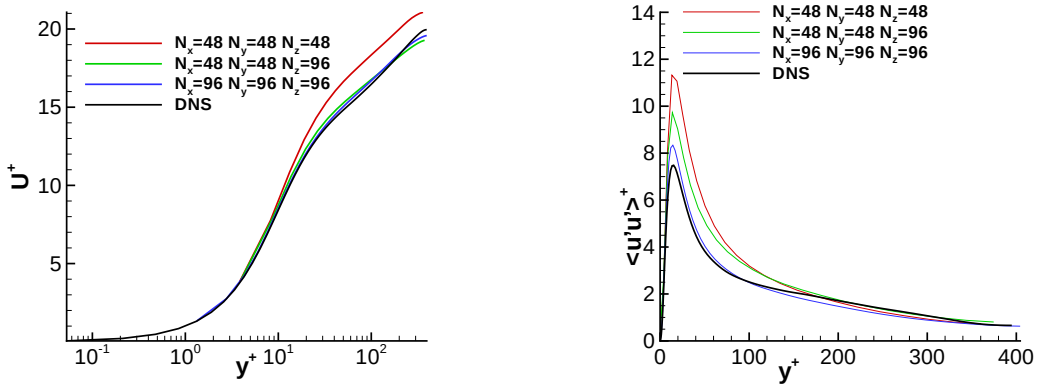

FIGURE 36: y^+ and U^+ (left) and fluctuations $\langle u'u' \rangle^+$ (right) for meshes with grid refinement in z-coordinate direction.

Now we will consider the mesh refinement in spanwise z-coordinate direction (Fig. 5.6). Reference grid is equal to the 48^3 mesh. Using the 48x48x96 mesh gives a reduction by a factor 4 in number of nodes and hence in computing time compared to the 96^3 mesh. Obviously a grid refinement in z-coordinate direction does improve the solution significantly. The quality of the solution on a mesh with $48 \times 48 \times 96$ nodes is almost as reliable as the solution we obtained by running the computation on a mesh with $96 \times 96 \times 96$ nodes.
As the cost of a computation on a grid which is refined in z-direction is much smaller than on a grid with refinement in all directions, it is advisable to refine the mesh in z-coordinate direction.

5.1.4 Convergence behavior for varying wall normal streching

The stretching parameter γ is used while generating a mesh. It is part of the hyperbolic tangent function which stretches the grid near the wall in y-coordinate direction. The basic idea is to change the stretching parameter in order to obtain more or fewer points in the boundary layer. We will study the question of whether the number of nodes should be concentrated in the near-wall region or in the channel center. Therefore, we will generate grids with more points in the channel and fewer points in the near-wall region and vice versa. By enlarging the stretching parameter more points are generated in the near-wall region — by minimizing it the number of points in the boundary layer decreases. We generate several grids with different stretching parameters. For a $48 \times 48 \times 48$ mesh we use $\gamma \in \{1.5, 1.72, 2.0, 2.2\}$. Note that the distance of the first node off the wall is crucial whether to use wall functions or not. In the following table, the distance of the first off-wall node is displayed for a $48 \times 48 \times 48$ mesh and various γ.

TABLE 1: Distance of the first off-wall node for various stretching-parameters γ

γ	1.5	1.72	2.0	2.2
$y(1)$	0.013	0.009846	0.006624	0.004929

Remember from (4.36) that we do not need to use wall functions as $y(1) < 0.018$. Previous computations have shown that it is advisable to choose $y(1)$ between 0.004 and 0.01.

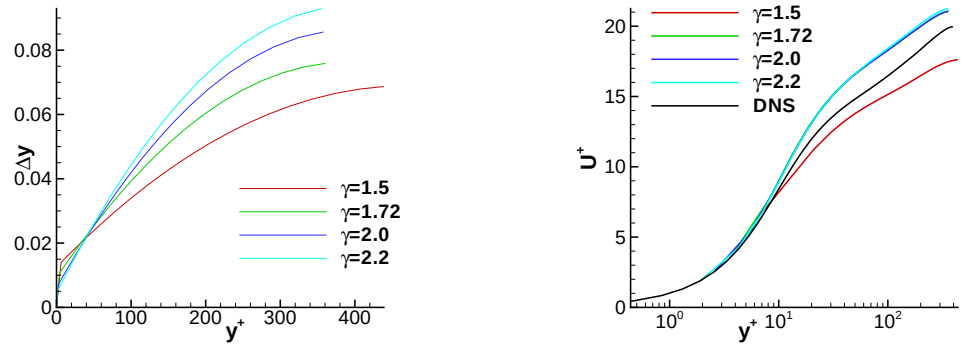

FIGURE 37: Δy (left) and the results of y^+ and U^+ (right) for various γ.

As γ increases, the number of points in the interior of the channel declines. Thus, Δy increases in direction of the channel center (Fig. 5.7 (left)). The results of U^+ and y^+ are similar for $\gamma = 1.72$, $\gamma = 2.0$ and $\gamma = 2.2$. For $\gamma = 1.5$ the result is worse in comparison to the DNS data (Fig. 5.7 (right)). Consequently, it is advisible to favor $y(1) < 0.01$ to obtain reliable results without wall functions.

5.1.5 Results for equidistant meshes using wall functions

In the previous computations we used meshes where Δy was computed via a hyperbolic tangent function in order to have a stretching near the wall. We will now

present the results from computations on an equidistant mesh with $48 \times 48 \times 48$ nodes. As the mesh width is equidistant, we obtain $\Delta y = \frac{2H}{48} \approx 0.04$. Note that the distance of the first off-wall node is too large for a computation of the friction velocity via the difference quotient (compare (4.36)). Therefore, we have to use the function of Reichardt to recover the friction velocity u_τ.

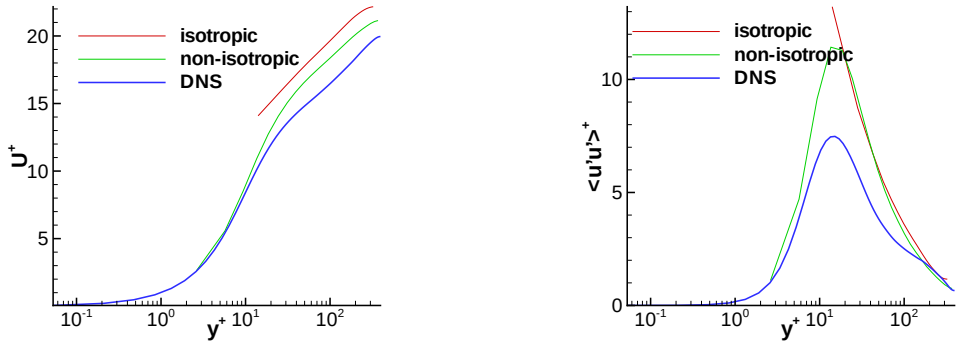

FIGURE 38: y^+ and U^+(left) and $\langle u'u' \rangle^+$ (right) for an equidistant mesh.

The approximation with the wall function supplies acceptable results. However, if we compare the data of U^+ and $\langle u'u' \rangle^+$ with the results of a computation on a non-isotropic grid where no wall functions were used, we see that the computation on the equidistant mesh using wall function is less precise.

5.2 Computation of the Kármán-sensor

Conceptually, there are two ways to evaluate the Kármán-sensor for turbulent flows. Firstly, the Kármán-sensor can be computed during the simulation for each time step. Thereafter the time averaged solution for the Kármán-sensor is calculated and during post-processing the sensor is averaged in the homogeneous streamwise and spanwise directions. The Kármán-sensor computed this way takes into account the velocity gradients in all three directions of all velocity components. Therefore, it will be denoted by $L_{vK,3D}$.

The second method can be used to evaluate the Kármán-sensor for the time- and space-averaged velocity field, where spatial averaging is performed over the homogeneous directions. Since, after averaging, the mean velocity profile is only a function of wall normal coordinate y, it will be denoted by $L_{vK,1D}$

$$L_{vK,1D} = \kappa \left| \frac{\partial \langle u \rangle / \partial y}{\partial^2 \langle u \rangle / \partial y^2} \right| .$$

The derivatives are computed with central differences according to section 3.2.3

$$\left(\frac{\partial u}{\partial y}\right)_i = \frac{u_{i+1} - u_{i-1}}{y_{i+1} - y_{i-1}}$$
$$\left(\frac{\partial^2 u}{\partial y^2}\right)_i = \frac{\left(\frac{\partial u}{\partial y}\right)_{i+\frac{1}{2}} - \left(\frac{\partial u}{\partial y}\right)_{i-\frac{1}{2}}}{y_{i+\frac{1}{2}} - y_{i-\frac{1}{2}}} = \frac{\frac{u_{i+1}-u_i}{y_{i+1}-y_i} - \frac{u_i-u_{i-1}}{y_i-y_{i-1}}}{\Delta y_i}$$

where y_i and u_i denote the values for y and u at the primary node P_i. Note that the mesh in y-coordinate direction is not equidistant. Therefore we have to compute the difference between y_{i-1} and y_{i+1}. The formula for the second order gradient is derived from the first order derivative between the virtual intermediate nodes $y_{i+\frac{1}{2}}$ and $y_{i-\frac{1}{2}}$. Δy_i will denote the difference between these nodes. This is in accordance with the length of the dual cell around the primary node P_i.

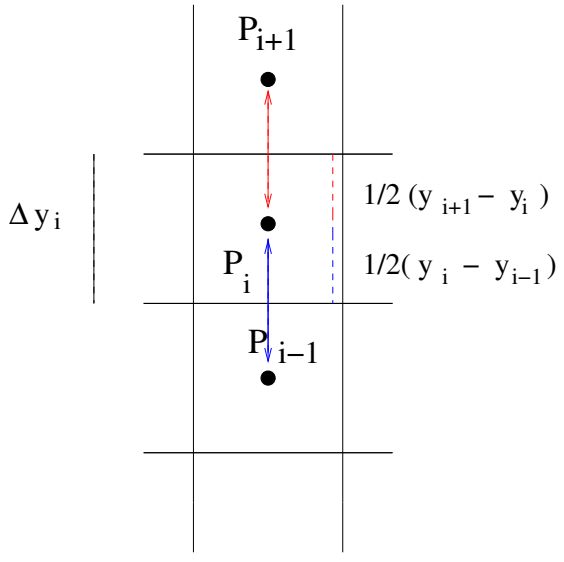

FIGURE 39: Length of the dual cell around P_i.

From the sketch we observe that the length of the dual cell Δy_i can be computed with the following formula:

$$\begin{aligned}\Delta y_i &= \frac{1}{2}(y_{i+1} - y_i) + \frac{1}{2}(y_i - y_{i-1}) \\ &= \frac{1}{2}(y_{i+1} - y_{i-1})\end{aligned}$$

Treatment of the boundary points
At the boundary points we need virtual points in order to compute the central differences. We will compute the central differences by treating the virtual points, y_{-1} and y_{N+1} as *mirror points*, i.e.,

$$\begin{aligned}u(y_{-1}) &= -u(y_1), \\ u(y_{N+1}) &= -u(y_{N-1}).\end{aligned}$$

For the gradients we obtain:

$$\begin{aligned}(\frac{\partial u}{\partial y})_{i=0} &= \frac{u_1 - u_{-1}}{y_1 - y_{-1}} = \frac{u_1 - (-u_1)}{2y_1} = \frac{u_1}{y_1}, \\ (\frac{\partial^2 u}{\partial y^2})_{i=0} &= \frac{u_1 - 2u_0 + u_{-1}}{\Delta y_0} = \frac{u_1 - 2u_0 - u_1}{2y_1} = 0,\end{aligned}$$

as we have the no-slip boundary condition at the wall.

5.3 Results of Kármán-sensor for mean velocity

For turbulent channel flow the mean velocity field is a function of wall normal coordinate y only. The relation $\nu_t = u_\tau \kappa y$ (4.17) is interpreted in the way that the boundary layer consists of "attached eddies" of velocity scale u_τ and length scale $l_{RANS} \equiv l_* = \kappa y$, see [28]. This model leads to the log law for the mean velocity. The quantity l_{RANS} is often referred to as RANS length scale. In the near-wall region, it is altered using the van Driest damping function. Thus L_{RANS} is defined by (see [21, p.304])

$$L_{RANS} = \kappa \cdot y \left[1 - \exp(-y^+/26)\right].$$

L_{RANS} may be computed from U^+ and y^+ from the LES simulation. Another approach is to compute $L_{vK,1D}$ from Reichardt's function

$$F(y^+) = \frac{\log(1 + 0.4y^+)}{\kappa} + 7.8\left[1 - \exp(-\frac{y^+}{11}) - \frac{y^+}{11}\exp(-\frac{y^+}{3})\right]$$

which gives an accurate approximation for the DNS data for U^+ and y^+. We use the values of y^+ of the DNS data of Moser, et al. [20] in order to compute $F(y^+) \approx U^+$. First and second order gradients are computed numerically with central differences and the one-dimensional Kármán-sensor is computed pointwise.

The third approach is to compute the gradients directly from the DNS data. This length scale is called $L_{vK,1D,DNS}$.

We plot the Kármán-sensor computed from the mean velocity field of the LES with THETA for several grids with N^3 nodes where $N = 24, 48, 64, 96$. The sensor is

compared to $L_{RANS}, L_{vK,Reichardt}$ and to $L_{vK,1D,DNS}$. The results are shown in Fig. 5.10 and Fig. 5.11 .

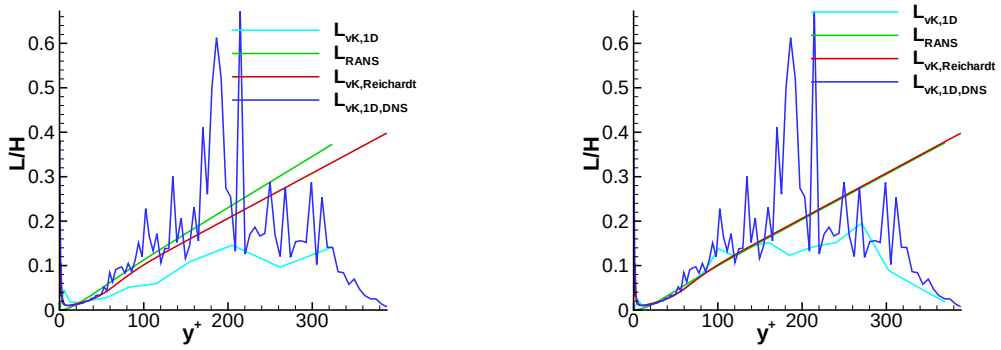

FIGURE 40: $L_{vK,1D}$ on a mesh with $N = 24$ (left) and $N = 48$ (right) in comparison to several length scales.

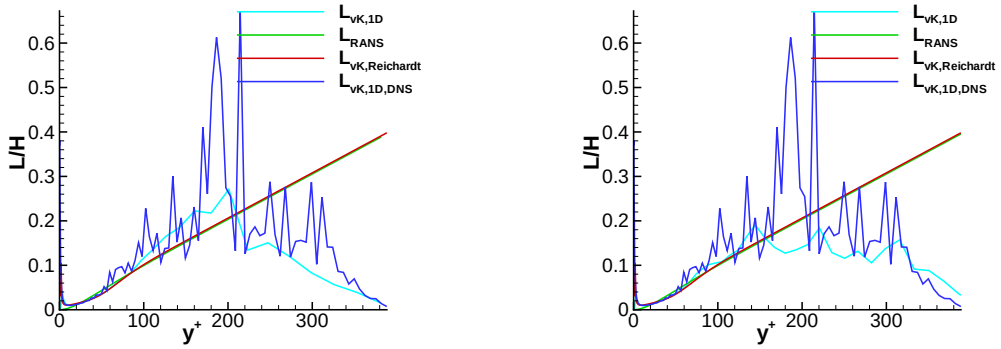

FIGURE 41: $L_{vK,1D}$ on a mesh with $N = 64$ (left) and $N = 96$ (right) in comparison to several length scales.

The agreement between L_{RANS} and $L_{vK,Reichardt}$ increases together with the grid refinement. Both length scales have a strong agreement for a mesh with 48^3 or more nodes. As stated in [21, p.304] L_{RANS} is an appropriate specification for the mixing length in the viscous sublayer and in the log-layer. $L_{vK,Reichardt}$ can be interpreted as the Kármán-sensor for mean velocity near the wall and in the log-law region. For turbulent channel flow at $Re_\tau = 395$, the log layer is in the region $40 \lessapprox y^+ \lessapprox 80$, which is around 20% of the channel half width (see [21, p.275]). It can be seen that in this region the agreement of $L_{vK,1D}$ and L_{RANS}, and therefore also $L_{Reichardt}$, increases with a grid refinement. L_{RANS},$L_{Reichardt}$ and $L_{vK,1D}$ almost perfectly agree for fine meshes with 64^3 or 96^3 nodes. Therefore, $L_{vK,1D}$ is able to detect the (RANS) length scale of the mean flow field.

5.4 Results of three-dimensional Kármán-sensor

In this section we will investigate the three-dimensional Kármán-sensor. The sensor should give information about the quality of the grid. Therefore, it is suitable to

compare the sensor to a grid parameter. In the previous sections we analyzed which grid parameters are important for an LES. Various grid spacing parameters will be compared and from the results of the previous sections we will choose a grid spacing parameter and compare it to the sensor. From the results we try to answer the second question, i.e. how the sensor could be used to estimate the quality of a grid.

5.4.1 Introduction and basic ideas

General behavior of the Kármán-sensor

In the first step we will plot the Kármán-sensor for the computations on various grids.

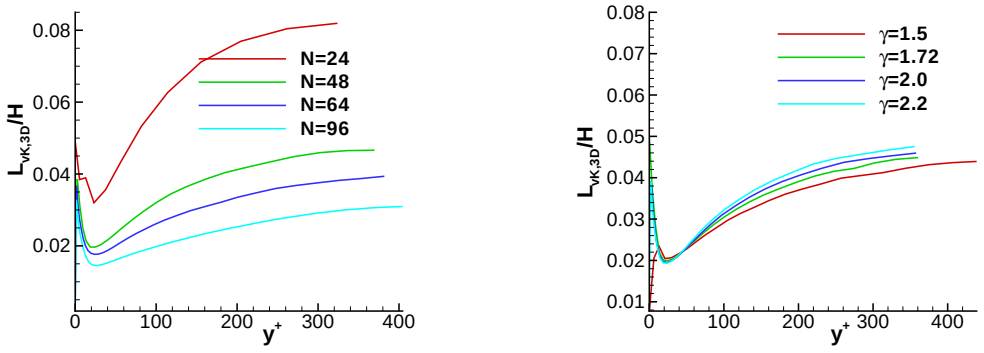

FIGURE 42: $L_{vK,3D}$ for grid refinement in all directions and γ=const. (left) and for $N = 48$ and various wall normal stretchings γ (right).

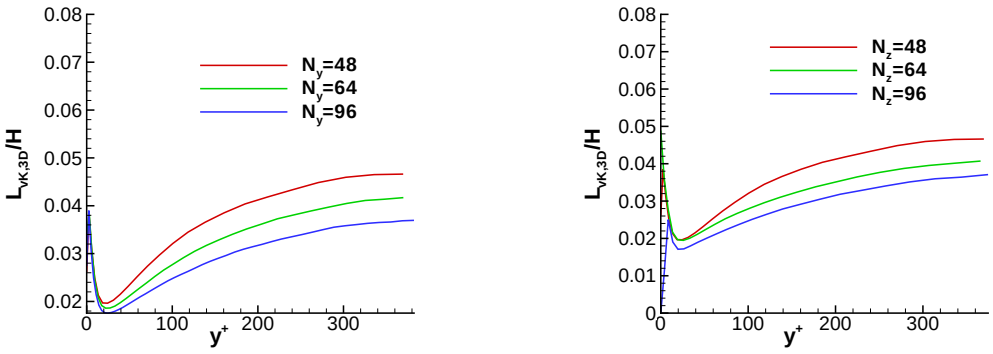

FIGURE 43: $L_{vK,3D}$ for grid refinement in y-coordinate direction and $N_x = N_z = 48$ (left) and refinement in z-coordinate direction and $N_x = N_y = 48$ (right).

For isotropic and non-isotropic grid refinement the sensor tends to become smaller with an increasing grid refinement (Fig. 5.12 (left) and Fig. 5.13). A reason for this could be that finer grids resolve smaller scales. As the sensor should indicate the length of the scales, it becomes smaller with grid refinement. For variation of the wall normal stretching the sensor does not change significantly (Fig. 5.12 (right)).

The goal of using a sensor like the Kármán-sensor for LES is to gain information

about the quality of the mesh. Ideally, the sensor should determine whether the grid is fine enough for the computation or if the grid needs to be refined. If the interpretation is right that the sensor indicates the length of the small scales, then the comparison to a parameter which represents the grid spacing could determine whether or not the grid is fine enough to resolve the small scales. In the following we will try to establish which parameter of the mesh should be compared with the sensor. Thus, we will plot several measurements of the grid, Δy, Δz and the average diameter of a control volume $\sqrt[3]{\Delta x \Delta y \Delta z}$. The grid parameter Δx is not taken into account as it is larger than Δz for the grids used in the channel flow.

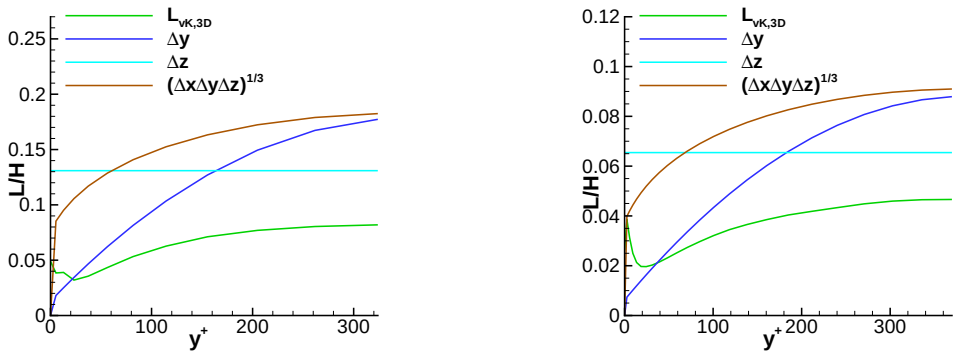

FIGURE 44: $L_{vK,3D}$ on a mesh with $N = 24$ (left) and $N = 48$ (right) in comparison to several length scales of the grid spacing.

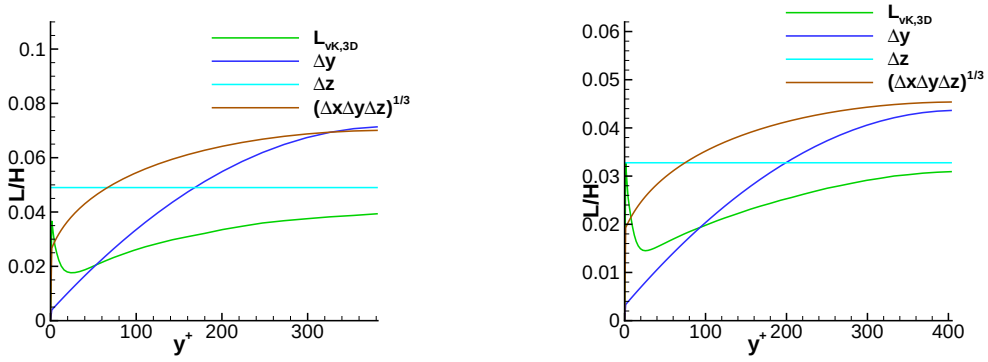

FIGURE 45: $L_{vK,3D}$ on a mesh with $N = 64$ (left) and $N = 96$ (right) in comparison to several length scales of the grid spacing.

Considering the results that are plotted in Fig. 5.14 and Fig. 5.15 we can state that the average diameter $\sqrt[3]{\Delta x \Delta y \Delta z}$ is always larger than Δy as Δx is constant and larger than Δy and Δz. The Kármán-sensor (3D) is smaller than Δz. For finer grids ($N = 96$) the sensor is almost as large as Δz. On the coarsest grid ($N = 24$) the sensor is much smaller than Δz. The grid parameter Δy is smaller than Δz near the wall, whereas in the interior of the channel Δz is smaller than Δy.

One hypothesis is:

Near the wall Δy is the dominating grid parameter on which the quality of the solution depends. In the interior of the channel Δz is the important grid parameter. In the next subsection we will attempt to review this hypothesis.

5.4.2 First approach for comparison of $L_{vK,3D}$ to a grid parameter

According to the hypothesis of the previous subsection it is advisible to compare the Kármán-sensor with Δy near the wall and with Δz in the interior of the channel. To combine Δy and Δz we take the minimum of both and hope to obtain a proper length scale for each part of the channel. We will denote the grid parameter with:

$$\Delta_{min} = \min(\Delta y, \Delta z).$$

To compare the Kármán-sensor with Δ_{min} we will compute the fraction of the sensor and Δ_{min}. Our goal is to obtain threshold values for ${}^{L_{vK,3D}}/_{\Delta_{min}}$ such that

Hypothesis H1

$$\begin{aligned}
{}^{L_{vK,3D}}/_{\Delta_{min}} < c_1 &\longrightarrow \text{The mesh is too coarse.}\\
{}^{L_{vK,3D}}/_{\Delta_{min}} > c_2 &\longrightarrow \text{The mesh is fine enough, where } c_2 < 1.\\
c_1 \leq {}^{L_{vK,3D}}/_{\Delta_{min}} \leq c_2 &\longrightarrow \text{It can not be determined whether the grid is fine enough.}
\end{aligned}$$

In the following we will analyze hypothesis H1. Therefore we consider the following questions:

- Which results support hypothesis H1?
- Which results contradict hypothesis H1?

The following figures (5.16-5.17) illustrate ${}^{L_{vK,3D}}/_{\Delta_{min}}$ for isotropic and non-isotropic grid refinement and for various stretching parameters γ.

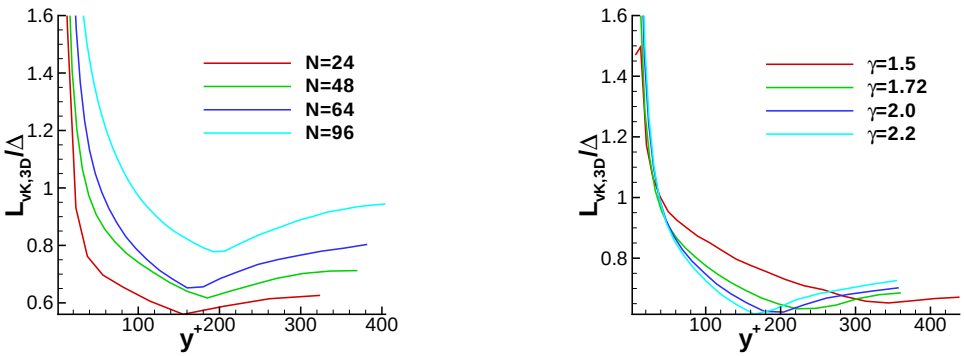

FIGURE 46: ${}^{L_{vK,3D}}/_{\Delta_{min}}$ for grid refinement in all directions and γ=const. (left) and for $N = 48$ and various wall normal stretching parameter γ (right).

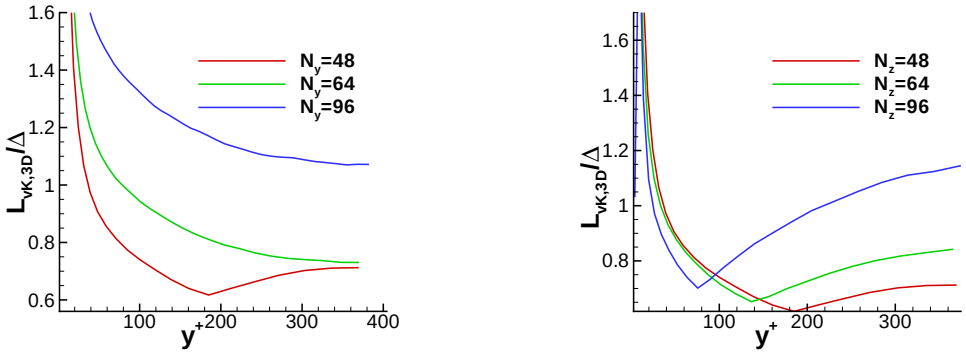

FIGURE 47: $L_{vK,3D}/\Delta_{min}$ for grid refinement in y-coordinate direction and $N_x = N_z = 48$ (left) and refinement in z-coordinate direction and $N_x = N_y = 48$ (right).

First we will consider the simulations that support hypothesis H1.

Therefore, we will study the results of the sensor for isotropic grid refinement. We observed in a previous section that the LES on a grid with $N = 96$ is sufficient. Figure 5.16 (left) shows that ${}^{L_{vK,3D}}/_{\Delta_{min}}$ approaches 1 as N increases. For $N = 24$ the fraction ${}^{L_{vK,3D}}/_{\Delta_{min}}$ is smaller and approaches a value of 0.6. Therefore, for the simulation with isotropic grid refinement, the sensor is able to indicate whether or not a simulation gives reliable results.
The results for grid refinement in z-coordinate direction in Figure 5.17 (right) also support hypothesis H1. The fraction ${}^{L_{vK,3D}}/_{\Delta_{min}}$ increases as the number of grid nodes in z-coordinate direction becomes larger. In the previous section we observed that a large number of nodes in z-coordinate direction improves the quality of the results significantly. Therefore, the tendency of ${}^{L_{vK,3D}}/_{\Delta_{min}}$ is in agreement with the quality of the results.

Now we will consider the results that contradict hypothesis H1.

We will consider meshes with a fixed number of nodes, i.e. 48^3 nodes and we will vary the point distribution in wall normal direction (Fig. 5.16 (right)). The sensor does not behave as expected from hypothesis H1. For $\gamma = 1.5$ the fraction ${}^{L_{vK,3D}}/_{\Delta_{min}}$ approaches 0.8. Hence, it is bigger than the value of ${}^{L_{vK,3D}}/_{\Delta_{min}}$ for $\gamma = 1.72, 2.0, 2.2$. In a previous section we observed that a simulation for $\gamma = 1.5$ does not give reliable results. This is a contradiction to H1.
In the next step we will study ${}^{L_{vK,3D}}/_{\Delta_{min}}$ on meshes with a refinement in y-coordinate direction. Again, the sensor does not behave as expected from H1. The results in Figure 5.17 (left) show that ${}^{L_{vK,3D}}/_{\Delta_{min}}$ increases as the mesh becomes finer in one coordinate direction. The value of ${}^{L_{vK,3D}}/_{\Delta_{min}}$ is larger than 1.2 for $N_y = 96$ which is larger than $c_2 < 1$ and larger than 0.8 for $N_y = 64$. In the previous section we observed that the results for the simulation on a $48 \times 64 \times 48$-mesh do not give reliable results. This contradicts hypothesis H1.

The problem of evaluating ${}^{L_{vK,3D}}/_{\Delta_{min}}$ for a grid refinement in one direction is that either $\min(\Delta z, \Delta y) = \Delta y$ or $\min(\Delta z, \Delta y) = \Delta z$ for all points of the grid. Consequently, the other grid parameter does not affect ${}^{L_{vK,3D}}/_{\Delta_{min}}$. For isotropic grid refinement Δ_{min} is a suitable grid parameter for comparison to $L_{vK,3D}$. If the grid is refined in one direction or if the stretching in y-coordinate direction changes it is not suitable to compare $L_{vK,3D}$ to Δ_{min}.
With regard to the contradictions to hypothesis H1 we can no longer hold this hypothesis.

5.4.3 Second approach for comparison of $L_{vK,3D}$ to a grid parameter

The problem of the first approach is that for non-isotropic grid refinement only Δz or Δy does affect ${}^{L_{vK,3D}}/_{\Delta_{min}}$. Our second approach is to use the square root of $\Delta y \Delta z$. We will denote our new grid parameter as

$$\Delta_{yz} := \sqrt{\Delta y \Delta z}.$$

Contrary to Δ_{min} the influence of Δy *and* Δz is guaranteed. From the fraction $L_{vK,3D}/\Delta_{yz}$ we hope to be able to determine whether the grid is fine enough for the simulation such that the following hypothesis holds.

Hypothesis H2

$$L_{vK,3D}/\Delta_{yz} < c_1 \longrightarrow \text{The mesh is too coarse.}$$
$$L_{vK,3D}/\Delta_{yz} > c_2 \longrightarrow \text{The mesh is fine enough, where } c_2 < 1$$
$$c_1 \leq L_{vK,3D}/\Delta_{yz} \leq c_2 \longrightarrow \text{It can not be determined whether the grid is fine enough.}$$

In the following figures (Fig. 5.18 and 5.19) the fraction of $L_{vK,3D}/\Delta_{yz}$ is plotted for isotropic and non-isotropic grid refinement and for various wall normal stretchings γ. With the results we will investigate if hypothesis H2 fits.

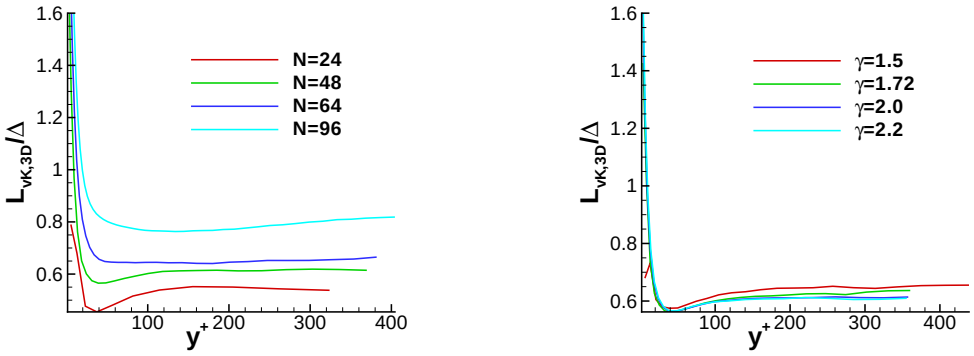

FIGURE 48: $L_{vK,3D}/\Delta_{yz}$ for grid refinement in all directions and γ=const. (left) and for $N = 48$ and various wall normal stretchings γ (right).

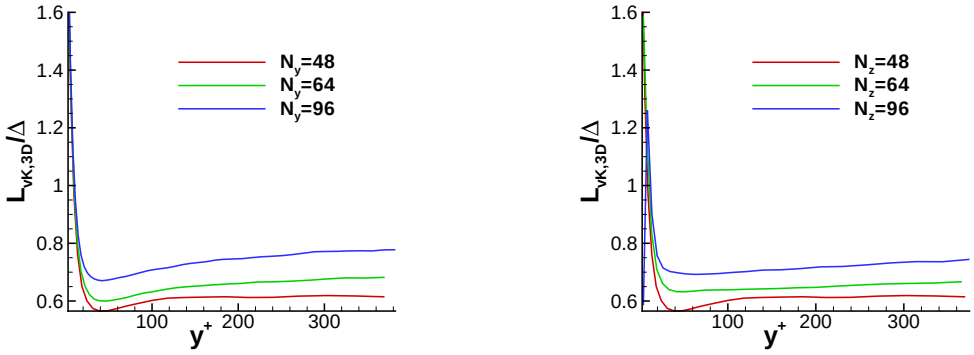

FIGURE 49: $L_{vK,3D}/\Delta_{yz}$ for grid refinement in y-coordinate direction and $N_x = N_z = 48$ (left) and refinement in z-coordinate direction and $N_x = N_y = 48$ (right).

The results of the various simulations mostly support hypothesis H2.

First we will consider the isotropic grid refinement (Fig. 5.18 (left)). For the finest grid with $N = 96$ the value of $L_{vK,3D}/\Delta_{yz}$ is approximately equal to 0.8. The values for the grids with mesh width $N = 48$ and $N = 64$ are between 0.5 and 0.8. For the

coarsest grid with $N = 24$ the value of ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ is approximately 0.5 and becomes even less than 0.5 near the wall. Thus the fraction ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ increases as the grid becomes finer. The tendency of ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ is in agreement with the numerical results.

In the second step we will consider the results for grid refinement in one direction (Fig. 5.19). We notice that the value of ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ is about 0.7 for grids with $N_y = 96$ or $N_z = 96$. For coarser grids the value is approximately 0.6 or even less. Hence the fraction ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ increases with a grid refinement in one coordinate direction but also takes into account the influence of the grid spacing in the non-refined coordinate direction. The tendency of the sensor is in agreement with the numerical results.

A crucial point regarding the sensor raises when comparing the sensor for the $48 \times 96 \times 48$ and the $48 \times 48 \times 96$ mesh. Obviously both sensors give a similar level, although for $N_z = 96$ the minimum is a little larger than for $N_y = 96$. In comparison $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \approx 0.7$ for $N_z = 96$ and $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \approx 0.65$ for $N_y = 96$. On the other hand, the prediction for mean velocity profiles is far better for $N_z = 96$ than for $N_y = 96$.

However the study of ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ needs some special care for meshes with different stretching parameter γ. If we take the sensor ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ into account for estimating the quality of the grid, the consequence is that the grid with $\gamma = 1.5$ would be the best for $N = 48$ as ${}^{L_{vK,3D}}/_{\Delta_{yz}}$ increases when γ becomes less (see Fig. 5.18 (right)). The reason for this fact is that the number of points in y-direction in the interior of the channel increases as γ becomes larger. Therefore Δy decreases in the interior of the channel. Consequently, the ratio of $L_{vK,3D}$ and $\sqrt{\Delta y \Delta z}$ becomes larger for a small stretching parameter γ. In the numerical results we observed that a computation for $\gamma = 1.5$ does not give accurate results for the channel flow.

We may say that hypothesis H2 is suitable for isotropic and partly suitable for non-isotropic grid refinement. The influence of the dominating grid parameters, Δy and Δz is guaranteed. For the variation of a stretching in wall normal direction the grid parameter is not suitable.

To summarize we will assume that the quality of a grid (with a proper wall normal stretching) is:

- reliable if $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \gtrapprox 0.8$
- of moderate quality if $0.6 \lessapprox \min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \lessapprox 0.8$
- not fine enough if $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \lessapprox 0.6$.

6. First results for turbulent flow over a backward-facing step

In this chapter we will present the first results with the WALE model and the Smagorinsky model of the flow over a backward-facing step, which has only one homogenous direction. Therefore, it is a statistically two-dimensional problem.

6.1 Test case setup

In this test case we will investigate the fluid motion over a cuboid domain, containing a step. In the following we will present how the test case is established.

- **Computational domain**
  The geometry is sketched in the following figure.

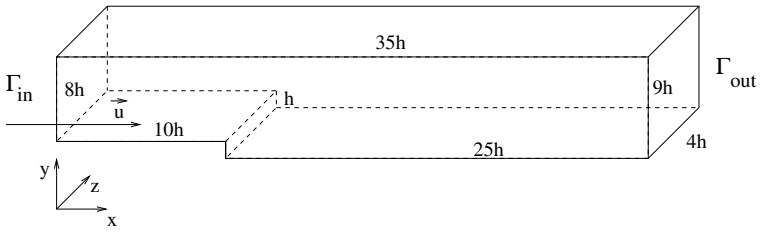

FIGURE 50: Backward-facing step.

  The step is located at $x = 0$ with step height $h = 0.0127\ m$. The computational domain consists of a streamwise length $L_x = 29h$, including an inlet section $L_{in} = 4h$, vertical height $L_y = 9h$ and spanwise width $L_z = 4h$.

- **Boundary conditions**
  At the xz-planes we prescribe a no-slip boundary condition on walls. In z-coordinate direction periodic boundary conditions are used. In streamwise x-coordinate direction we will prescribe inlet and outlet flow conditions.

- **Reynolds number**
  The *step-height Reynolds number* is defined as:

$$Re_h = \frac{U_0\ h}{\nu}$$

  where U_0 denotes the initial velocity and ν the kinematic viscosity chosen to be $1.5 \cdot 10^{-5}$. The Reynolds number we will consider is $Re_h = 37500$.

- **Inlet and Outlet flow conditions**
  A blending of DNS data in the near-wall region from Spalart [26] and experimental data from Driver and Seegmiller [6] is otherwise used for the mean velocity profile at the inflow boundary.

- **Computational grid**
  We will use different grids with $78 \times 31 \times 16$ and $110 \times 47 \times 32$ nodes in x-,y- and z-coordinate direction. In y-coordinate direction the grid is anisotropic with a fine mesh near the bottom wall generated by a hyperbolic tangent function (compare to (4.21)). In spanwise direction the mesh is equidistant and in streamwise direction it is mostly equidistant, only a small stretching in the region near the outlet exists.

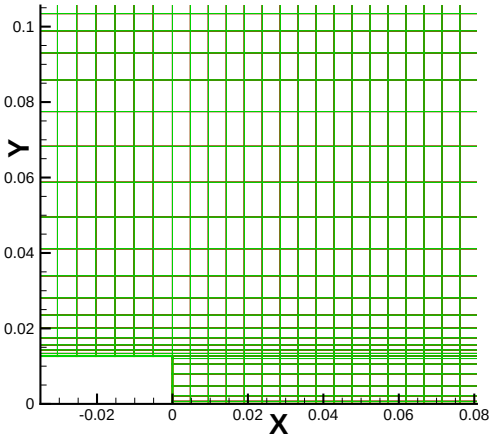

FIGURE 51: Mesh with $78 \times 31 \times 16$ nodes for backward-facing step.

- **Computational time step**
  We will run the computation with time step sizes $\Delta t = 2 \times 10^{-5} s$ and $\Delta t = 1 \times 10^{-5} s$. One computation is performed over $0.2s$ which is analogous to 10000 and 20000 time steps, respectively.

- **Initial Conditions**
  We will initialize the velocity field with the inflow centerline velocity $U_0 = 44.2 \frac{m}{s}$ for the velocity in streamwise direction. The velocity component in y- and z-coordinate direction is set to zero.

- **Turbulence Model**
  We will use the same turbulence model as already used in the channel flow, i.e.

large-eddy simulation with WALE model. Additional simulations were performed using the *Smagorinsky model*.

- **Wall function**
  Wall functions are used at the upper and bottom walls. Therefore relatively coarse grids in the near-wall region can be used.

- **Statistical Averaging**
  The statistical averaging is performed over a simulation time $\frac{348h}{U_0} \approx 0.1s$ after a flow developing time $\frac{348h}{U_0} \approx 0.1s$ which is around 12 "flow-through" times.

- **Wall skin friction coefficient**
  The wall skin friction velocity C_f is denoted by:

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho U_0^2},$$

  where τ_w denotes the shear stress on the wall. The length from the step to the point where C_f turns from negative to positive is the reattachment length. In the experiment of Driver and Seegmiller [6] the reattachment length is 6.48 h.

> "[...] The location of reattachment was inferred from a linear interpolation of the oil-flow laser skin-friction measurements.[...]"

  Since wall functions are used, the wall skin friction coefficient C_f is computed using the wall function of Reichardt (4.34). The averaged C_f is computed from the statistical and spatial averaged velocity field during postprocessing.

- **Postprocessing for the backward-facing step**
  Unlike the channel flow, the backward-facing step has only one homogeneous direction which is the spanwise direction. Consequently, a spatial averaging over all points for which the x- and y-coordinate are equal has to be applied. According to the time and spatial filtering in the previous chapter we will apply for test case backward-facing step the following formula

$$\langle \vec{u} \rangle \approx \langle \langle \vec{u} \rangle_s \rangle_t = \frac{1}{N_z(T_{stop} - T_{start})} \sum_{k=1}^{N_z} \left(\sum_{l=T_{start}}^{T_{stop}} \vec{u}(x, y, z_k, t_l) \right),$$

  where N_z denotes the number of points in z-coordinate direction. The spatial averaging is sketched in the following figure.

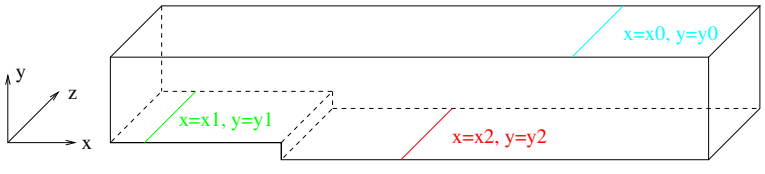

FIGURE 52: Spatial averaging over all points for which x- and y-coordinate are equal.

6.2 Smagorinsky model

The Smagorinsky model had been proposed by Smagorinsky (1963) [24]. Similar to the WALE model it is an eddy-viscosity model. In the Smagorinsky model the eddy viscosity is denoted by:

$$\nu_t = l_S^2 \left|S\right| = \left(C_S \Delta\right)^2 \left|S\right|,$$

with $|S| = (2\mathbb{S}(\vec{U}) : \mathbb{S}(\vec{U}))^{\frac{1}{2}}$ the characteristic rate-of-strain for mean velocity field $\vec{U}$. The scalar l_s denotes the Smagorinsky length scale which is proportional to the filter width Δ with the Smagorinsky contant C_S. In our computations we have used a Smagorinsky constant $C_S = 0.1$ and $C_S = 0.05$. The filter width Δ is often taken as the cubic root of the cell volumes, $\Delta = V_c^{\frac{1}{3}}$ with V_c being the volume of control volumes.

6.3 Numerical results

In the following sections we will present the first numerical results for the velocity field and the Kármán-sensor. The results for the WALE model and the Smagorinsky model, using two different Smagorinsky constants, are shown for two time step sizes. The results were successfully computed on the meshes with $78 \times 31 \times 16$ and $110 \times 47 \times 32$ nodes, whereas on the grid with $169 \times 71 \times 32$ nodes no results could be obtained as the computation using THETA did not converge.

6.3.1 Results of flow using WALE model

For the backward-facing step we used a WALE constant of $C_W = 0.5$. The computations were run for time step size $\Delta t = 2 \times 10^{-5}\ s$ on the meshes with $78 \times 31 \times 16$ and $110 \times 47 \times 32$ nodes. In the following the instantaneous velocity $\vec{u}$ is plotted for time $t = 0.2s$ for a computation on a mesh with $110 \times 47 \times 32$ nodes. In contrast to [12] no synthetic turbulence is added to the mean velocity at the inlet boundary. Therefore the formulation of 3D turbulent vortices in the shear-layer is slower than in [12].

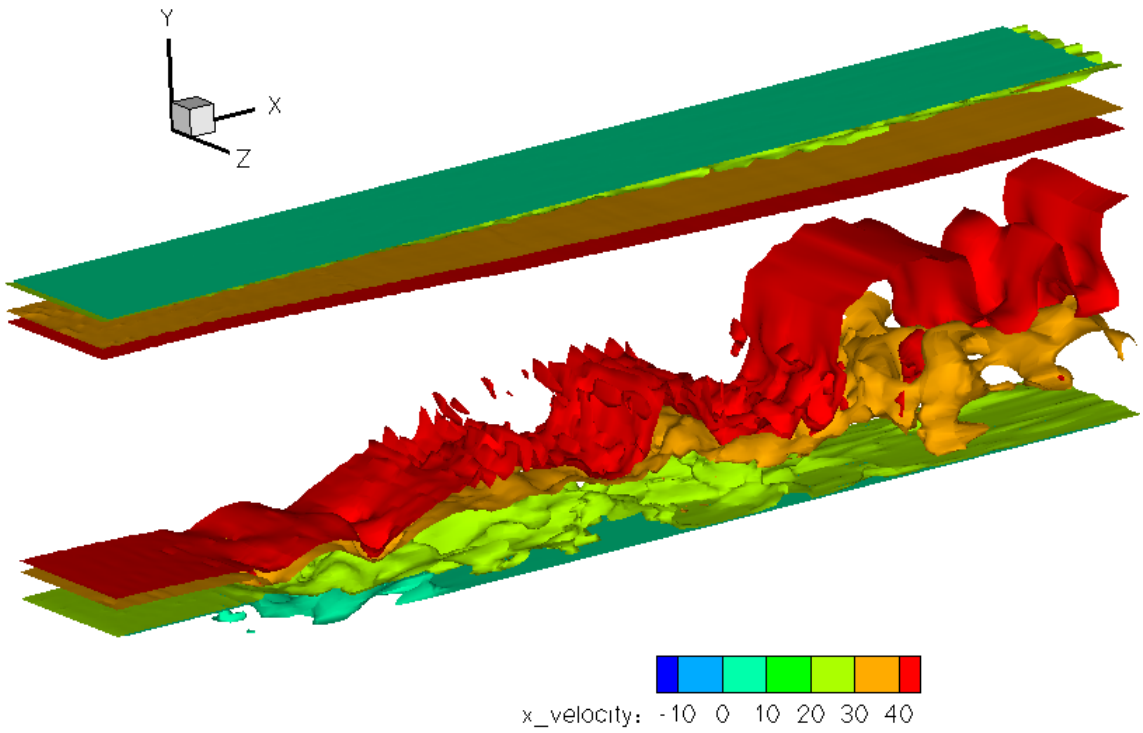

FIGURE 53: Isosurfaces of the instantaneous velocity.

The spatial and time averaged velocity field depends on x- and y-coordinate. We plot the mean velocity $\vec{U}$, fluctuations $\langle u'u' \rangle$ and $\langle v'v' \rangle$ depending on the y-coordinate for various, discrete values for x. The results are compared to the experimental data of Driver and Seegmiller [6]. In the following the results for mean velocity field and fluctuations are shown for the computations on both grids and time step size $\Delta t = 2 \times 10^{-5} s$.

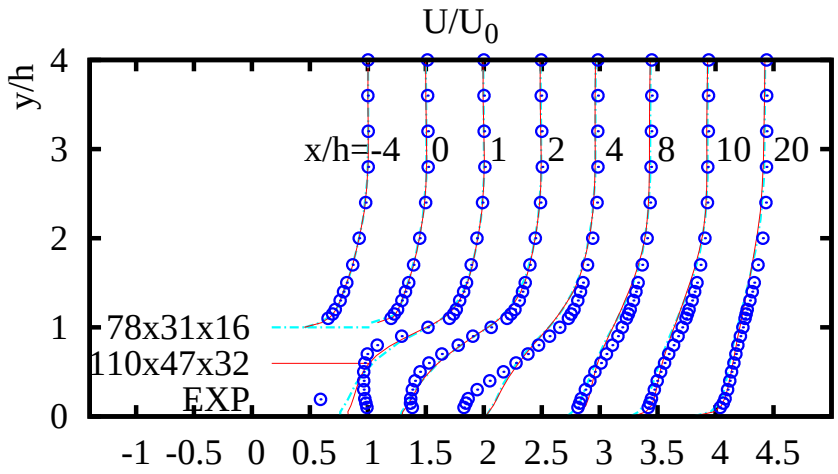

FIGURE 54: Mean streamwise velocity for various meshes; ○:experimental data.

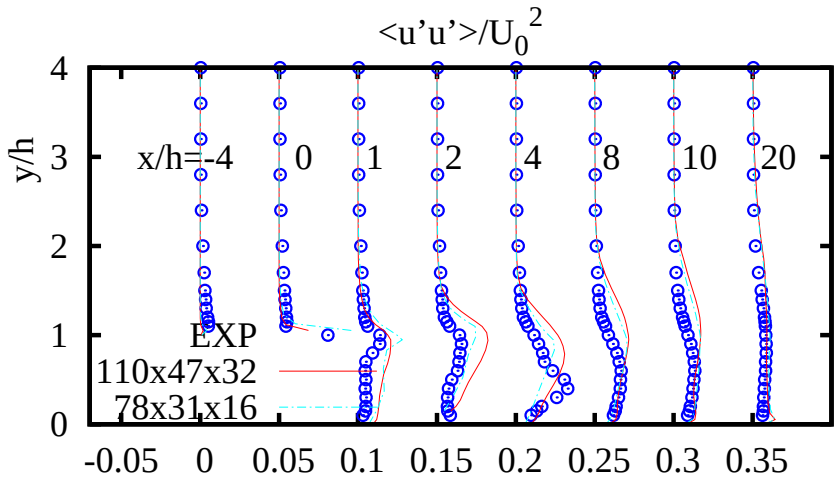

FIGURE 55: Fluctuations $\langle u'u' \rangle$ for various meshes; ○:experimental data.

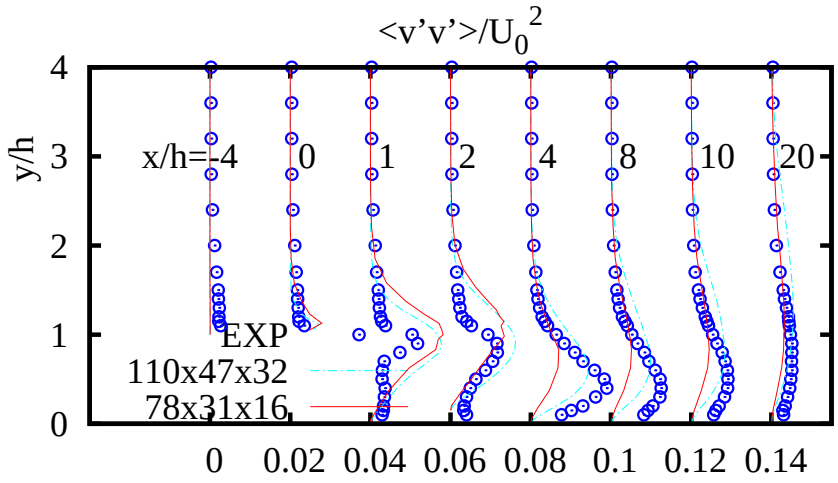

FIGURE 56: Fluctuations $\langle v'v' \rangle$ for various meshes; ○:experimental data.

The results for the computations with the WALE model concur with the experimental data. However, it is not obvious that the results on the fine mesh are better than on the coarse mesh. The fluctuations are large between $x = h$ and $x = 4h$, i.e. near the step. We can therefore conclude that the flow is turbulent around the step. The skin friction velocity C_f turns from negative to positive at $x \approx 3.56h$ for the computation on the grid with $78 \times 31 \times 16$ nodes and at $x \approx 3.75h$ for the computation on the $110 \times 47 \times 32$-mesh.

6.3.2 Results of flow using Smagorinsky model

We also computed the flow over a backward-facing step using the Smagorinsky model, with a Smagorinsky constant $C_S = 0.05$ and $C_S = 0.1$. As before, the computations were run on meshes with $78 \times 31 \times 16$ and $110 \times 47 \times 32$ nodes using two time step sizes $\Delta t = 1 \times 10^{-5} s$ and $\Delta t = 2 \times 10^{-5} s$.

Results for Smagorinsky model with $C_S = 0.05$

In the following figures the results for the mean flow velocity and for the fluctuations are presented for time step size $\Delta t = 1 \times 10^{-5}$ and $C_S = 0.05$.

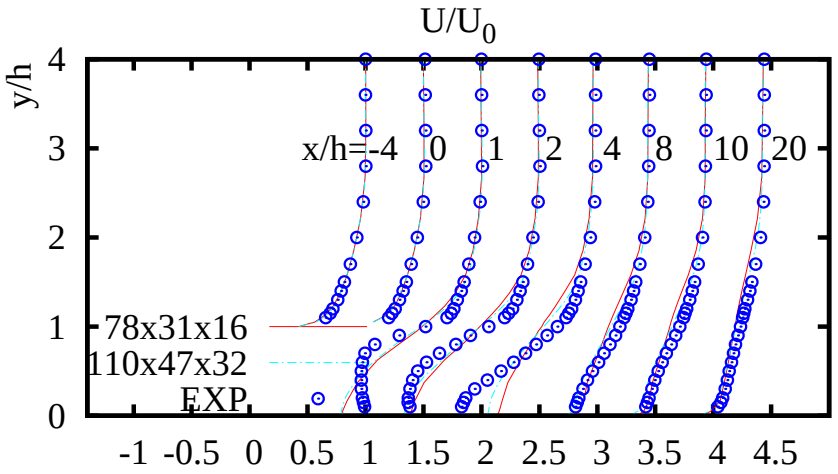

FIGURE 57: Mean streamwise velocity for various meshes; ∘:experimental data.

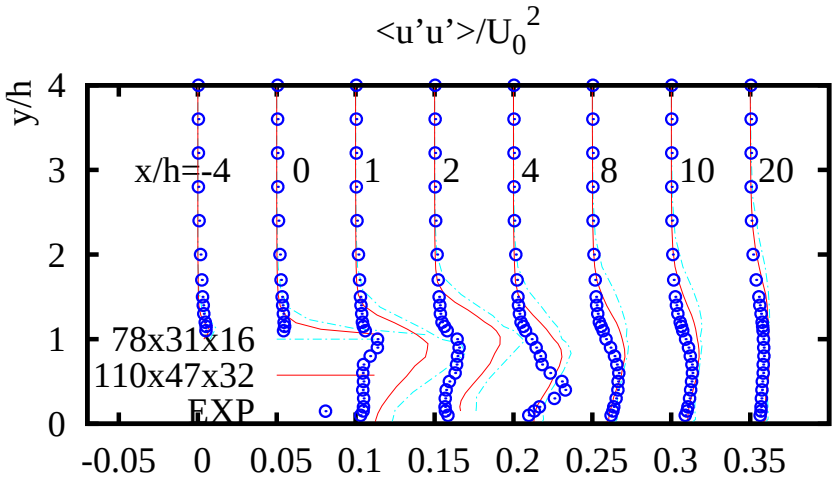

FIGURE 58: Fluctuations $\langle u'u' \rangle$ for various meshes; ∘:experimental data.

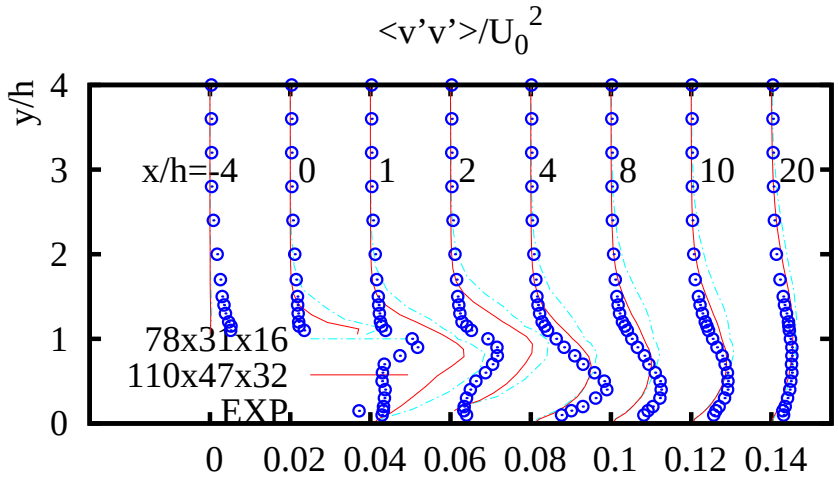

FIGURE 59: Fluctuations $\langle v'v' \rangle$ for various meshes; ∘:experimental data.

The results obtained by the computation with $C_S = 0.05$ are acceptable but do not exactly concur with the experimental data. With regard to the results of the fluctuations we observe that the solution differs a lot from the experimental data. For the reattachment length we obtained $x \approx 2.44h$ for the coarse mesh; and $x \approx 3.32h$ for the finer mesh.

Results for Smagorinsky model with $C_S = 0.1$

For computation with a Smagorinsky constant $C_S = 0.1$ and $\Delta t = 1 \times 10^{-5} s$ we obtained the following results for the mean velocity in streamwise direction and for the fluctuations.

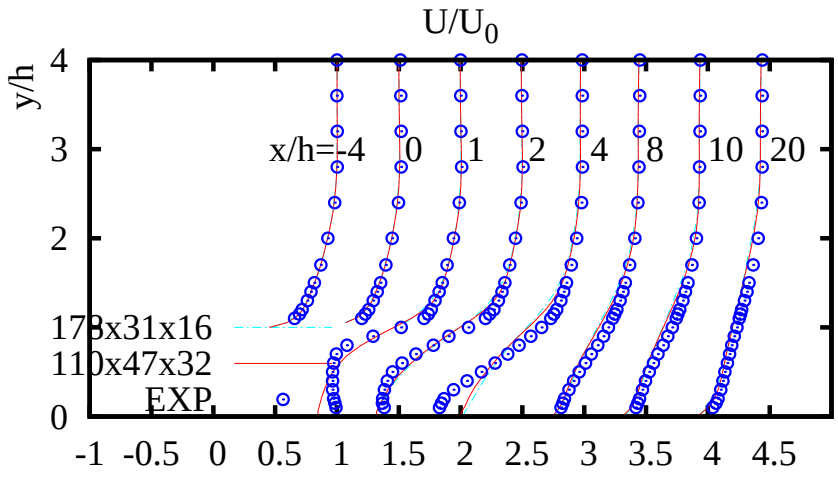

FIGURE 60: Mean streamwise velocity for various meshes; ∘:experimental data.

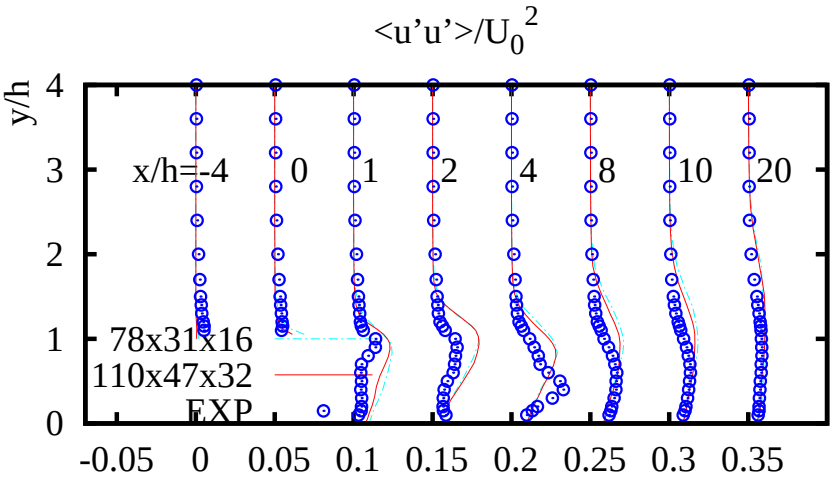

FIGURE 61: Fluctuations $\langle u'u' \rangle$ for various meshes; ∘:experimental data.

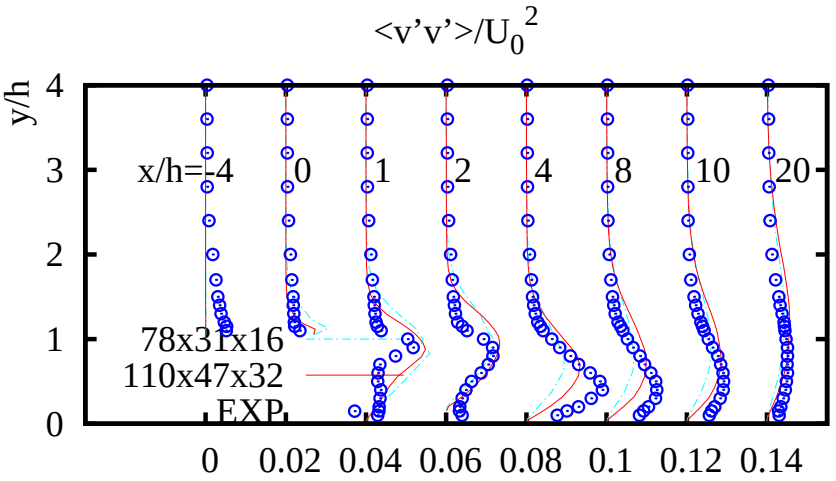

FIGURE 62: Fluctuations $\langle v'v' \rangle$ for various meshes; ∘:experimental data.

For the computations with the Smagorinsky constant $C_S = 0.1$ the results for the mean flow and the fluctuations are more reliable than the solution we obtained using $C_S = 0.05$. The results concur with the experimental data. The reattachment length is $x \approx 3.56h$ for the coarse mesh; and $x \approx 3.96h$ for the fine mesh.

In the following the skin friction coefficient is plotted for computations with WALE model and Smagorinsky.

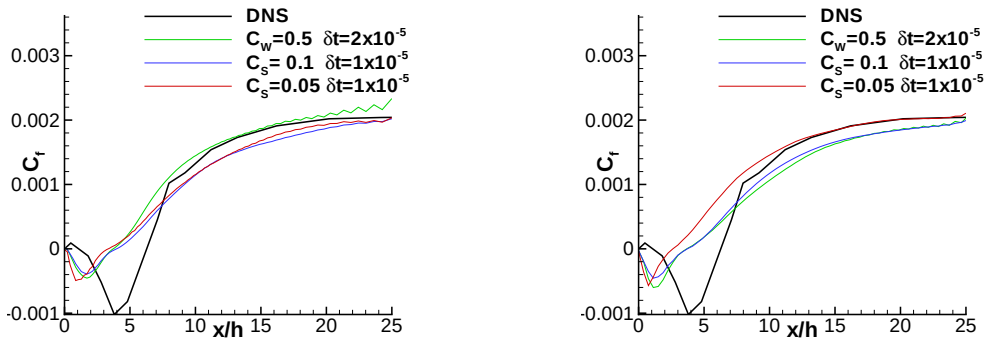

FIGURE 63: Skin friction velocity for various computations on a mesh with $110 \times 47 \times 32$ nodes (left) and with $78 \times 31 \times 16$ nodes (right).

The experimental data correspond best of all with the reattachment length for the computation using the Smagorinsky model with $C_S = 0.1$ as the length is approximately at $x = 3.96h$ and has therefore the smallest distance to the experimental solution $x = 6.48h$.

N.B. 6.3.1. *One important reason for the deviation of the solution for the reattachment length is that, contrary to the computation in the experiments of [6], no turbulent fluctuations at the inlet boundary are used.*

6.3.3 Results of Kármán-sensor for backward-facing step

During the simulation with THETA we computed the time averaged Kármán-sensor. In the post-process the spatial averaged sensor is computed. In general we can divide the domain into two regions:

- region 1: In the channel center we obtain for the velocity $\vec{u} \approx U_0 \vec{e}_x = const$ and $\langle u_i' u_j' \rangle \approx 0$. Therefore, the application of the sensor is not useful.

- region 2: In the shear layer and recirculation area near the step the fluctuations $\langle u_i' u_j' \rangle$, gradients and the Laplacian are large. Therefore, the application of the sensor is useful.

We will plot the sensor and the velocity vector (u, v) for $z = const$ on the xy-plane. Hereby we will focus on the region near the step. We obtain the following first results for the sensor.

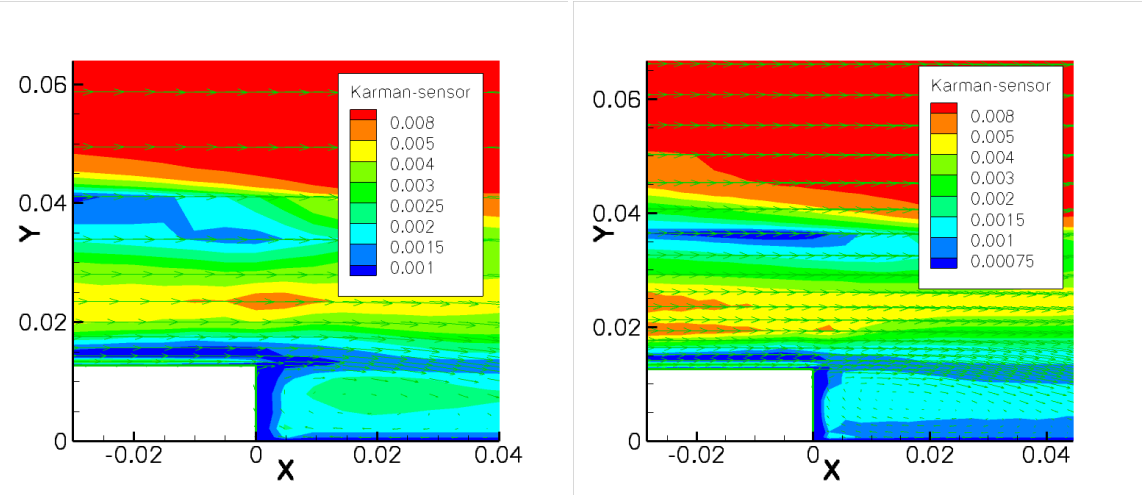

FIGURE 64: Kármán-sensor and velocity vector for WALE model on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

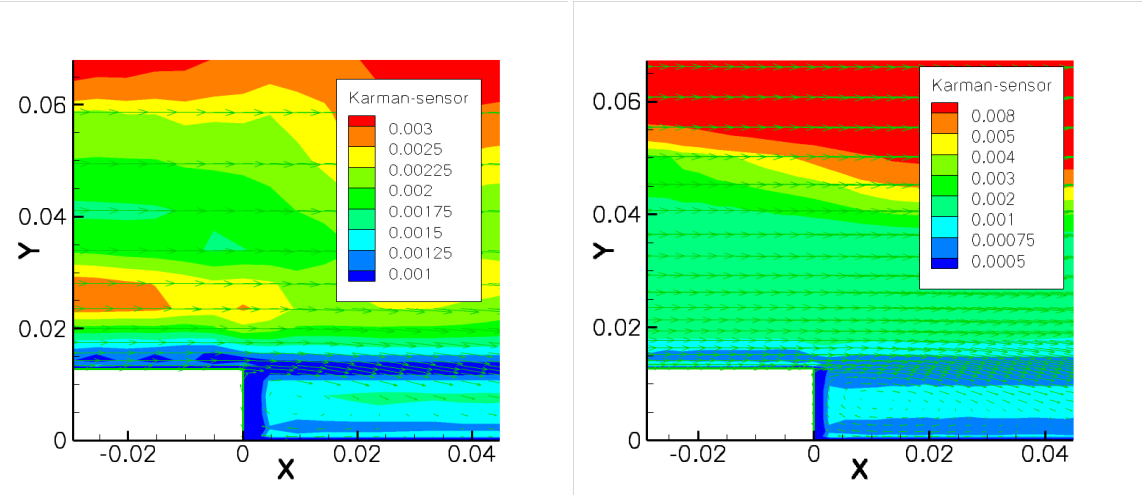

FIGURE 65: Kármán-sensor and velocity vector for Smagorinsky model with $C_S = 0.05$ on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

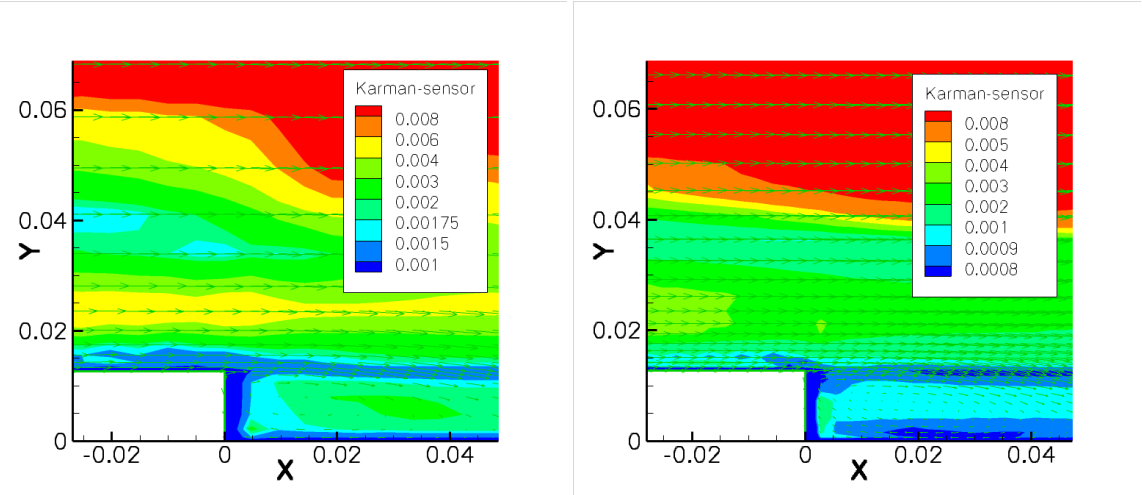

FIGURE 66: Kármán-sensor and velocity vector for Smagorinsky model with $C_S = 0.1$ on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

In the middle of the channel the velocity is almost uniform $\vec{u} = U_0 \vec{e}_x$. Therefore, the gradient and the Laplacian are almost zero. Moreover, the turbulent fluctuations are almost zero in the channel core. Therefore, the definition of L_{vK} would need to be corrected to treat such regions of almost uniform flow. We will not consider this region in any more detail since the Kármán-sensor is not designed for this region.

In the region behind the step the Kármán-sensor is small. After the step we have large velocity gradients. Due to the recirculation region near the bottom wall the second order velocity derivatives also enlarge. Moreover, the fluctuations are very large in the region of the shear layer and the recirculation region. It is thus of interest to study the Kármán-sensor in this region.

In the previous chapter we determined that the sensor has to be compared to a grid parameter. The grid parameter Δ_{yz} seemed to fit best. Therefore, in this case, we compare the sensor with Δ_{yz} as well. The results are presented in the following figures:

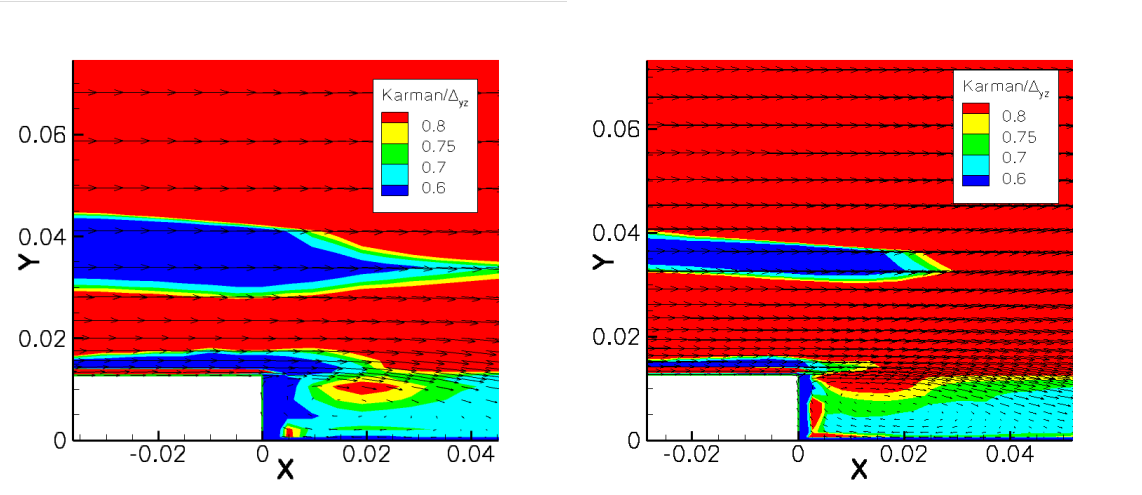

FIGURE 67: L_{vK}/Δ_{yz} and velocity vector for WALE model on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

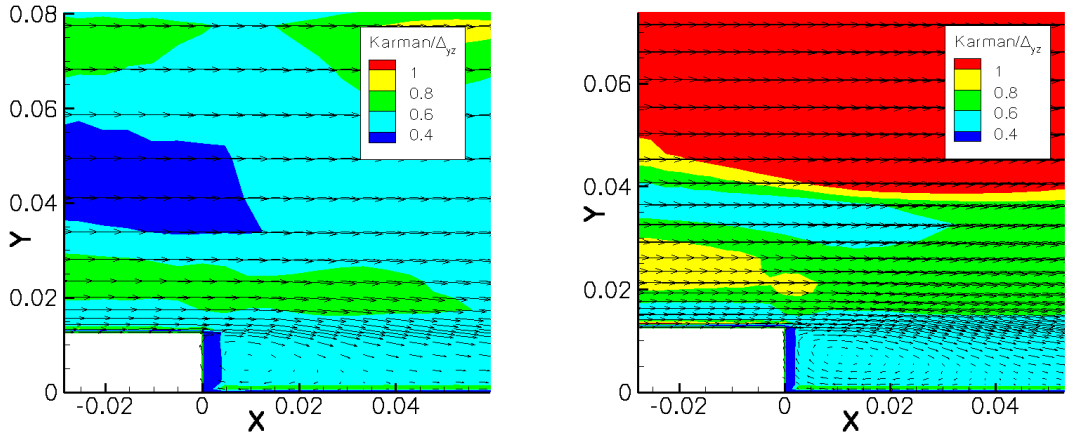

FIGURE 68: $^{L_{vK}}/_{\Delta_{yz}}$ and velocity vector for Smagorinsky model with $C_S = 0.05$ on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

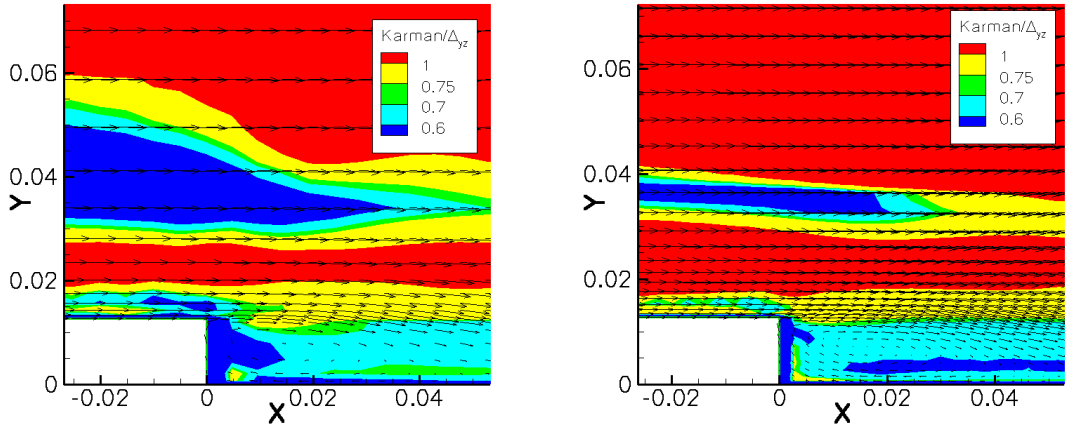

FIGURE 69: $^{L_{vK}}/_{\Delta_{yz}}$ and velocity vector for Smagorinsky model with $C_S = 0.1$ on a $78 \times 31 \times 16$-mesh (left) and on a $110 \times 47 \times 32$-mesh (right).

In the region behind the step the sensor divided by Δ_{yz} has values between 0.6 and 0.8 for the computation with the WALE model. The values are similar on both grids. For the computation with the Smagorinsky model using the constant $C_S = 0.05$, $^{L_{vK}}/_{\Delta_{yz}}$ has values between 0.4 and 0.6 for both grids. For the computation with $C_S = 0.1$ the sensor divided by Δ_{yz} has values between 0.6 and 0.8. It is stated that the mesh with $169 \times 71 \times 32$ nodes is fine enough for the computation of the flow over a backward-facing step but the meshes with $78 \times 31 \times 16$ nodes and $110 \times 47 \times 32$ nodes are not fine enough. As stated before, the computations on the mesh with $169 \times 71 \times 32$ nodes could not be completed successfully.

In the previous chapter we stated that the mesh was fine enough if $^{L_{vK}}/_{\Delta_{yz}}$ has values larger than 0.8. This is not the case for the computations using the WALE model and Smagorinsky with $C_S = 0.1$. The values for the computation with a Smagorinsky constant of 0.05 are smaller than 0.6 in the major part of the area right behind the step. If we follow the hypothesis of the previous chapter the results obtained by the WALE model with $C_W = 0.5$ and the Smagorinsky model with $C_S = 0.1$ were acceptable but not outstanding whereas the results of the computation with a

Smagorinsky constant $C_S = 0.05$ were defective.

If we compare the tendency of the sensor with the numerical results of the mean flow and the fluctuations we can say that the quality of the results are in harmony. The most defective results were those for the computation using Smagorinsky and $C_S = 0.05$ whereas the results were acceptable for the other two computations.

7. Conclusions

In this chapter we will summarize the main results of this thesis. Thereafter some open questions will be addressed.

7.1 Summary

In this thesis we considered modelling and simulation for turbulent flows and the corresponding numerical methods. As a guideline, the following questions were posed in the introduction:

- How can the motion of a fluid be described?
- How can the motion of a fluid be computed?
- How can turbulent flows be modeled?

We observed in Chapter 2 that the incompressible Navier-Stokes equations describe the motion, i.e. velocity, $\vec{u}$ and pressure, p of an incompressible fluid within a certain time interval $[0, T]$ and a closed domain, Ω.

The difficulty in solving these coupled equations using a projection method, i.e. to find $\vec{u}$ and p such that the equations are fulfilled in $[0, T] \times \Omega$ was described in Chapter 2. In order to illustrate the emerging problem of stability for the pressure, we considered the projection method for a structured scheme using the paper of [7]. We described the numerical schemes used in the DLR-THETA-code for simulation of incompressible viscous fluid flow problems. THETA is the unstructured flow solver for flows with small compressibility effects developed at DLR. It employs an unstructured second-order finite volume method on hybrid meshes. It is based on the dual mesh approach. Decoupling of velocity and pressure is accomplished using a projection scheme. The discretization of the fluxes is also prescribed there.

In Chapter 4, we considered the question how to solve turbulent flows. The problem in solving the smallest scales was discussed. In this thesis the LES model is the governing turbulence model. Hereby the effect of the small scales onto the large

scales is modeled such that the large scales can be resolved properly. Four conceptual points of an LES have been introduced. For the test case channel flow we discussed how to solve the Navier-Stokes equations using LES WALE model; how to compute the wall shear stress; and how to average the velocity field in order to compute a solution that represents the fluid motion within the whole time interval and domain. The treatment of viscous walls for turbulent flows was also described in detail.

In the following we will consider the question of numerical resolution for turbulent channel flow at $Re_\tau = 395$. A proper resolution is one of the key issues for LES.

- What are the resolution requirements for time step size and grid design when using the DLR-THETA-code, which is of second order accuracy in space and time?
- What are the requirements for statistical convergence?
- How can the efficiency of computations be improved by using a mesh with optimal resolution properties in streamwise, spanwise and wall normal direction?

In Chapter 5 we investigated a proper adjustment for the time averaging and the grid spacing concerning the channel flow. Considering a time step size $\Delta t = 0.17s$ the statistical averaging at time interval $[8000\Delta t, 16000\Delta t]$ supplied accurate results. Reasonable convergence behavior was observed. Concerning the grid spacing we obtained acceptable results for a simulation on a grid with 48^3 nodes and accurate results for a mesh with 96^3 nodes.

We also studied how the solution changes when the wall normal stretching is varied. The result is that the first mesh node $y(1)$ is chosen to be $y(1) \leq 0.01$ corresponding to $y^+(1) \leq 3.95$ and $y_\delta^+ \leq 0.9$ for the shifted node. For a larger value for $y(1)$ we obtained, without using wall functions, defective results. The simulation was also computed on an equidistant mesh using wall functions. The results were acceptable but we have seen that the simulation on an anisotropic mesh without using wall functions gives more accurate results.

Investigating the question which coordinate direction has the largest influence on the quality of the result of an LES simulation for the channel flow, we discovered that the z-coordinate direction has the largest influence on the quality. Therefore, the efficiency of a computation could be perfected by refining the grid in z-coordinate direction. For comparison, recall that the resolution requirements for a DNS are: $\Delta x^+ = 7.5$, $\Delta y^+ = 0.03$ and $\Delta z^+ = 4.4$ for the test case of turbulent channel flow by Moser [20] and $\Delta x^+ = 14.3$, $\Delta z^+ = 4.8$ for test case of a turbulent boundary layer by Spalart [26]; for details see [19]. However, recent results showed that the resolution by Spalart is not satisfactory. Results by Moser state that $\Delta x^+/\Delta z^+ \approx 2$. For the LES on the 64^3 mesh with $\Delta x^+ = 38.8$ and $\Delta z^+ = 19.4$ a similar ratio of $\Delta x^+/\Delta z^+$ is obtained, but the nodes are not distributed efficiently between x- and z-direction. On the $48 \times 48 \times 96$-mesh, agreement with the DNS results are much better than for the 64^3 mesh. On the $48 \times 48 \times 96$-mesh, we obtain: $\Delta x^+ \approx 52$ and $\Delta z^+ \approx 13$, i.e. $\Delta x^+/\Delta z^+ \approx 4$.

Regarding the main scientific questions, the focus of this thesis was to investigate the resolution requirements for the spatial discretization for LES. The aim was to

assess the resolution quality a posteriori, i.e. after a simulation, using so-called single grid estimators. Such a single grid estimator (also called sensor) reads a mean flow solution averaged in time and over homogeneous directions and then returns a scalar variable which is larger than a certain threshold value in regions where the resolution is fine enough and a value which is smaller than a certain threshold value in regions where the mesh needs to be refined. In Knopp, et al. [12] a sensor based on the resolved turbulent kinetic energy was considered. In this thesis we considered a sensor based on the so-called von Kármán length scale. The following questions were discussed:

- What is the Kármán-sensor?
- How can the Kármán-sensor be computed?
- How does the sensor behave in a fluid motion?
- How is it possible to assess the quality of a simulation of turbulent flow using the Kármán-sensor?

A derivation of the formula of the Kármán-sensor is given in Appendix B. For the computation of the sensor the first and second order gradients are necessary. In the THETA-code Green-Gauß and Least Square methods are implemented. In Chapter 3, the quality of these numerical methods for computing gradients was compared to the computation using central differences. Therefore, these methods were applied to the test cases "Sinus 1D" and "Sinus 2D". Then the Kármán-sensor was computed for test case "Sinus 2D". The results of these precursor studies are as follows:

- There is not much of a difference between the quality of the recovered gradients which were computed by using the Green-Gauß and the Least Square method. By using central differences for the computation we obtained an error which has been smaller by a factor 4 due to the smaller stencil.
- The computation of the Kármán-sensor for test case "Sinus 2D" showed that relative changes of the flow structure, due to different values for ω, can be easily detected by the sensor. Regarding the absolute values of the sensor, we obtained values which are smaller by a factor 4 than the corresponding characteristic vortex diameter.

In Chapter 5 and Chapter 6 the Kármán sensor was applied to the benchmark test cases channel flow and flow over a backward-facing step.

Let us first review the results for the channel flow. We will consider the simulations on isotropically refined meshes. Firstly, the *absolute level* of the predictions for the Kármán-sensor was considered. We observed that L_{vk} decreases when the mesh is isotropically refined. This showed that L_{vk} is linked at least somehow to the size of the turbulent structures resolved by the mesh. Secondly, the *relative level* of L_{vk} to a proper grid scale was investigated. The first task was to find a proper grid scale. The question was whether or not the ratio of L_{vk} and a proper grid scale can be used as an indicator to assess the resolution quality of the mesh.

Regarding the question of a proper grid scale we took into account the conclusion that the z-coordinate direction has the largest influence on the quality of an LES simulation for the channel flow. Concurrently, the y-coordinate direction is important for the LES simulation on the channel flow as it is the wall normal direction. We worked out two grid spacing parameters which were compared to the sensor. The result: the most fitting grid spacing parameter for the channel flow is $\Delta_{yz} = \sqrt{\Delta y \Delta z}$. A large value for ${}^{L_{vK}}/_{\Delta_{yz}}$ means a high quality whereas a small value indicates that the quality of the computation is defective. In our hypothesis we declared upper and lower bounds which are connected to the value of ${}^{L_{vK}}/_{\Delta_{yz}}$ and hence to the quality of the simulation. For the upper and lower bounds we suggest that the quality of an LES simulation on a channel flow is as follows

- reliable if $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \gtrapprox 0.8$
- of moderate quality if $0.6 \lessapprox \min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \lessapprox 0.8$
- defective if $\min({}^{L_{vK,3D}}/_{\Delta_{yz}}) \lessapprox 0.6$.

As a result, we have obtained together with the grid spacing parameter Δ_{yz} a sensor which increases monotonically if the grid is refined isotropically for the test case of turbulent channel flow.

Secondly, we studied how the sensor behaves for anisotropically refined meshes. Consequently, we may say that the application of sensor ${}^{L_{vK}}/_{\Delta_{yz}}$ is limited. With an excessive refinement in wall normal direction the results of the LES simulation do not improve but the sensor indicates a better quality.

We also examined how the sensor behaves if the wall normal stretching is changed. The result: the sensor is unsuitable to indicate whether or not the wall normal stretching is chosen properly. The sensor indicated a high quality for a simulation on a mesh with 48^3 nodes and a stretching parameter $\gamma = 1.5$ but the numerical results were not accurate for this simulation.

In Chapter 6 we analyzed how the sensor behaves for the flow over a backward-facing step. Due to stability problems of the THETA-code, only results on coarse and moderately fine grids could be obtained. The results are as follows:

The absolute values of the Kármán-sensor are small in regions where the mesh is fine. In the free shear-layer behind the step where small flow structures exist the sensor takes small values.

For comparing the sensor to Δ_{yz} we considered only the region behind the step. Similar to the channel flow we observed a correlation between the values of the sensor and the quality of the results. Computation using the WALE model with $C_W = 0.5$ and the Smagorinsky model with $C_S = 0.1$ supplied acceptable results. The sensor takes values between 0.6 and 0.8, similarly to the values for the channel flow and supports the findings that the meshes used are not fine enough. Computation using the Smagorinsky model and $C_S = 0.05$ supplied defective results. In this case the sensor takes values between 0.4 and 0.6. An increasing sensor, when going from the coarse mesh to a moderately fine mesh, as it was for the channel flow, indicates that the trend of the mesh quality is reasonably predicted. Note that the study for the flow over a backward-facing step is not finished as LES on fine meshes could not be successfully computed.

7.2 Open Questions

There are still open questions that could be studied in the future, such as:

- The interpretation of the Kármán-sensor for flow over a backward-facing step could not be completed as it was not possible to obtain results of the backward-facing step flow on the fine mesh with $169 \times 71 \times 32$ nodes. In addition, due to the lack of time, we were not able to investigate which grid spacing design was the most suitable and efficient for the backward-facing step.

- The Kármán-sensor should be applied to more complex test cases in order to investigate in greater depth the hypothesis of this thesis. One or two test cases are not enough for giving a well validated result, in particular, bearing in mind the goal is to apply the sensor to industrial problems.

- After a thorough investigation of the sensor, it is reasonable to apply the sensor in order to improve the efficiency of a computation, e.g. by using the sensor for grid adaptation or for the hybrid RANS-LES model.

- The role of the numerical method used for computing the gradients and second derivatives in L_{vK} needs to be investigated further. How sensitive are the levels of L_{vK} compared to the mesh spacing with respect to the numerical method?

- It would be interesting to investigate whether or not L_{vK} can be used as a length scale for the resolved scales in the subgrid scale models (WALE model or Smagorinsky model), instead of the grid spacing h used in standard LES. This addresses a possible link between LES and RANS simulations as developed by Menter for the SAS model.
  We should take note that $L_{vK,1D}$ is close to the RANS length scale for attached boundary layer flow. In this case, the eddy viscosity is given by Prandtl's mixing length formula which is the starting point of virtually every RANS model.
  The aim of such a blending between RANS and LES using a sensor based on the flow field is to avoid LES mode when the mesh is too coarse. As we observed for channel flow and backward-facing step, a too coarse grid for an LES gives results which are far less accurate than those of RANS.

A. Krylov subspace methods

Krylov subspace method

A Krylov subspace method is a projection method used to solve non-symmetric linear equation systems. K_m is the *Krylov subspace*

$$K_m = K_m(A, \vec{r}_0) = \text{span}\{\vec{r}_0, A\vec{r}_0, \ldots, A^{m-1}\vec{r}_0\}$$

and the tansposed Krylov subspace is defined as

$$K_m^T = K_m(A^T, \vec{r}_0) = \text{span}\{\vec{r}_0, A^T\vec{r}_0, \ldots, A^{Tm-1}\vec{r}_0\}$$

where $\vec{r}_0 = \vec{b} - A\vec{x}_0$ is the 0th residue and $\vec{x}_0$ the starting vector.
In Krylov subspace methods the jth approximation $\vec{x}_j$ is computed such that it is orthogonal to a j-dimensional space L_j. L_j is the Krylov subspace or a variation from this.
With the biconjugate gradients stabilized (BiCGSTAB) method van der Vorst [29] introduced a variant of the conjugate gradients stabilized (CGS) method from Sonneveld [25]. The advantage of BICGSTAB is that it converges much more smoothly. It uses different polynoms for the definition of searching directions and residue vectors. To understand BiCGSTAB, it is important to be familiar with the ideas of biconjugate gradients (BiCG) and CGS method. The most important results are as follows:

For the jth iterated vector $\vec{x}_j$ the condition

$$\vec{b} - A\vec{x}_j \perp K_j^T = \vec{r}_j \perp K_j^T = \{\vec{r}_0, A^T\vec{r}_0, \ldots, (A^T)^{j-1}\vec{r}_0\}$$

holds. From this it follows that $\vec{r}_j \perp K_m^T$ for $m \leq j$.
Consequently, for all polynoms $\pi_{j-1} \in \mathcal{P}_{j-1}$ it follows that $(\vec{r}_j, \pi_{j-1}(A^T)\vec{r}_0) = 0$.
The residue vectors $\vec{r}_j$, $\vec{r}_j^*$ and the searching vectors $\vec{p}_j$, $\vec{p}_j^*$ are defined according to CGS method.

$$\vec{r_j} = \phi_j(A)\vec{r}_0 \tag{A.1}$$
$$\vec{r_j^*} = \phi_j(A^T)\vec{r}_0 \tag{A.2}$$
$$\vec{p_j} = \psi_j(A)\vec{r}_0 \tag{A.3}$$
$$\vec{p_j^*} = \psi_j(A^T)\vec{r}_0 \tag{A.4}$$

The polynoms ϕ_j and ψ_j are defined recursively as:

$$\begin{aligned}\phi_0(\lambda) &= \psi_0(\lambda) = 1\\ \psi_j(\lambda) &= \phi_j(\lambda) + \beta_{j-1}\psi_{j-1}(\lambda)\\ \phi_{j+1}(\lambda) &= \phi_j(\lambda) - \alpha_j\lambda\psi_j(\lambda)\end{aligned}$$

Furthermore we define

$$\begin{aligned}\tilde{p}_j^* &:= \tilde{r}_j^* := \phi(A^T)\vec{r}_0\\ \phi_{j+1}(\lambda) &:= (1-\omega_j\lambda)\phi_j(\lambda)\phi_0 := 1.\end{aligned}$$

It follows that

$$\phi_j(A)\vec{r}_0 = \vec{r}_j \perp K_j^T$$

and therefore

$$(\phi_j(A)\vec{r}_0, \pi_{j-1}(A^T)\vec{r}_0)_2 = 0$$

for all $\pi_{j-1} \in \mathcal{P}_{j-1}$. From this fact and with the recursive definitions of the polynoms in (A.1) we obtain

$$\begin{aligned}\alpha_j &= \alpha_j \frac{(\varphi_j(A)\vec{r}_0, \phi_j(A^T)\vec{r}_0)_2}{(\varphi_j(A)\vec{r}_0, \phi_j(A^T)\vec{r}_0)_2}\\ &= \frac{(\phi_j(A)\varphi_j(A)\vec{r}_0, \vec{r}_0)_2}{(\frac{\varphi_j(A)-\varphi_{j+1}(A)}{\alpha_j}\vec{r}_0, \phi_j(A^T)\vec{r}_0)_2}\\ &= \frac{(\phi_j(A)\varphi_j(A)\vec{r}_0, \vec{r}_0)_2}{(\phi_j(A)A\psi_j(A)\vec{r}_0, \vec{r}_0)_2}.\end{aligned}$$

Therefore we define

$$\begin{aligned}\hat{r_j} &:= \phi_j(A)\varphi_j(A)\vec{r}_0\\ \hat{p_j} &:= \phi_j(A)\psi_j(A)\vec{r}_0.\end{aligned}$$

We define $\hat{s_j}$ and follow

$$\begin{aligned}\hat{s}_j &:= \phi_j(A)\varphi_{j+1}(A)\vec{r}_0\\ &= \phi_j(A)\varphi_j(A)\vec{r}_0 - \alpha_j A\phi_j(A)\psi_j(A)\vec{r}_0\\ &= \hat{r}_j - \alpha_j A\hat{p}_j.\end{aligned}$$

So we obtain the recursive formula

$$\begin{aligned}\hat{r}_{j+1} &= \phi_{j+1}(A)\varphi_{j+1}(A)\vec{r}_0\\ &= (\mathbb{I} - \omega_j A)\phi_j(A)\varphi_{j+1}(A)\vec{r}_0\\ &= (\mathbb{I} - \omega_j A)\hat{s}_j\end{aligned}$$

and

$$\begin{aligned}\hat{p}_{j+1} &= \phi_{j+1}(A)\psi_{j+1}(A)\vec{r}_0 \\ &= \hat{r}_{j+1} + \beta_j(\mathbb{I} - \omega_j A)\hat{p}_j.\end{aligned}$$

From the recursive definitions of the polynoms we obtain

$$\varphi_j(\lambda) = (-1)^j \alpha_{j-1} \dots \alpha_0 \lambda^j + \overline{\pi}_{j-1}(\lambda).$$

And similarly

$$\phi_j(\lambda) = (-1)^j \omega_{j-1} \dots \omega_0 \lambda^j + \hat{\pi}_{j-1}(\lambda)$$

with $\overline{\pi}_{j-1}(\lambda), \hat{\pi}_{j-1}(\lambda) \in \mathcal{P}_{j-1}$. With $\vec{r}_j^* = \phi_j(A^T)\vec{r}_0$ we obtain the equation

$$\begin{aligned}\frac{(\vec{r}_j, \vec{r}_j^*)_2}{(\vec{r}_j, \tilde{\vec{r}}_j^*)_2} &= \frac{(\varphi_j(A)\vec{r}_0, \varphi_j(A^T)\vec{r}_0)_2}{(\varphi_j(A)\vec{r}_0, \phi_j(A^T)\vec{r}_0)_2} \\ &= \frac{\alpha_{j-1} \dots \alpha_0}{\omega_j \dots \omega_0}.\end{aligned}$$

So we conclude for β_j

$$\beta_j = \frac{(\vec{r}_{j+1}, \vec{r}_{j+1}^*)_2}{(\vec{r}_j, \vec{r}_j^*)_2} = \frac{\alpha_j}{\omega_j} \frac{(\hat{r}_{j+1}, \vec{r}_0)_2}{(\hat{r}_j, \vec{r}_0)_2}.$$

With ω_j we have obtained a parameter for minimization of the residue. Let $f_j : \mathbb{R} \to \mathbb{R}$ be the function defined as follows:

$$f_j(\omega) := ||(\mathcal{I} - \omega A)\hat{s}_j||_2^2$$

For $\hat{s}_j \neq 0$ we have : $f_j''(\omega) = 2||A\hat{s}_j||_2^2 \rangle 0$. In this case the function is strictly convex. We will constrain on this case and obtain

$$\omega_j := \underbrace{\arg \min f_j(\omega)}_{\omega \in \mathbb{R}}$$

with

$$\begin{aligned}0 = f_j'(\omega_j) &= -2(A\hat{s}_j, \hat{s}_j)_2 + 2\omega_j(A\hat{s}_j, A\hat{s}_j)_2 \\ \text{and} \quad \omega_j &= \frac{(A\hat{s}_j, \hat{s}_j)_2}{(A\hat{s}_j, A\hat{s}_j)_2}.\end{aligned}$$

The BICGSTAB iterated approximation can be written as follows:

$$\begin{aligned}\hat{x}_{j+1} &= A^{-1}(\vec{b} - \hat{r}_{j+1}) \\ &= A^{-1}(\vec{b} - \hat{s}_j + \omega_j A\hat{s}_j) \\ &= A^{-1}(\vec{b} - \hat{r}_j + \alpha_j A\hat{p}_j + \omega_j A\hat{s}_j) \\ &= \hat{x}_j + \alpha_j \hat{p}_j + \omega_j \hat{s}_j\end{aligned}$$

Neglecting the superscript $\hat{}$ we obtain with these conclusions the BiCGSTAB algorithm in the following form.

BiCGSTAB-algorithm
Given a linear equation system $A \cdot \vec{x} = \vec{b}$:

Choose $\vec{x}_0 \in \mathbb{R}^n$ and $\epsilon \geq 0$ and compute:

1. $\vec{x}_0 \in \mathbb{R}^n \quad$ and $\quad \epsilon > 0$
2. $\vec{r}_0 := \vec{p}_0 := \vec{b} - A\vec{x}_0$, $\rho_0 := (\vec{r}_0, \vec{r}_0)_2$, $j := 0$

While $\quad ||\vec{r}_j||_2 > \epsilon$

3. $\vec{v}_j := A\vec{p}_j$, $\alpha_j := \dfrac{\rho_j}{(\vec{v}_j, \vec{r}_0)_2}$
4. $\vec{s}_j := \vec{r}_j - \alpha_j \vec{v}_j$, $\vec{t}_j := A\vec{s}_j$
5. $\omega_j := \dfrac{(\vec{t}_j, \vec{s}_j)_2}{(\vec{t}_j, \vec{t}_j)_2}$
6. $\vec{x}_{j+1} := \vec{x}_j + \alpha\vec{p}_j + \omega_j \vec{s}_j$
7. $\vec{r}_{j+1} := \vec{s}_j - \omega_j \vec{t}_j$
8. $\rho_{j+1} := (\vec{r}_{j+1}, \vec{r}_0)_2$, $\beta_j := \dfrac{\alpha_j}{\omega_j} \dfrac{\rho_{j+1}}{\rho_j}$
9. $\vec{p}_{j+1} := \vec{r}_{j+1} + \beta_j(\vec{p}_j - \omega_j \vec{v}_j)$, $j := j + 1$

For more detailed explanations see [17].

B. Kármán length scale

According to the work of Menter and Egorov [18], we will consider a shear flow with mean flow in streamwise x-direction (with direction unit vector $\vec{e}_1$). We will assume that the mean velocity profile U only depends on the y-coordinate direction (with direction unit vector $\vec{e}_2$). Then the velocity field is of the form:

$$\vec{u}(x,y,z,t) \;=\; U(y)\vec{e}_1 \;+\; \vec{u}'(x,y,z,t). \tag{B.1}$$

The autocovariance function of $\vec{u}'$, also called two-point correlation, is defined in [21] as

$$R_{ij}(\vec{r},\vec{x},t) \;=\; \langle u_i'(\vec{x},t)u_j'(\vec{x}+\vec{r},t)\rangle. \tag{B.2}$$

The autocovariance function $R_{ij}(\vec{r},\vec{x},t)$ gives informations on the spatial structure of the fluctuating velocity field. In the case of the shear flow (B.1), Menter and Egorov are interested in:

$$\sum_{i=1}^{d} R_{ii}(r_y\vec{e}_2,\vec{x},t) \;=\; \langle \vec{u}'(\vec{x},t)\cdot\vec{u}'(\vec{x}+r_y\vec{e}_2,t)\rangle. \tag{B.3}$$

Note that from the definition of turbulent kinetic energy k we obtain:

$$2k \;=\; \sum_{i=1}^{d} R_{ii}(0,\vec{x},t) \;=\; \langle \vec{u}'(\vec{x},t)\cdot\vec{u}'(\vec{x},t)\rangle.$$

The quantity

$$\frac{1}{k}\int_{-\infty}^{\infty}\sum_{i=1}^{d} R_{ii}(r_y\vec{e}_2,\vec{x},t)\mathrm{d}r_y$$

has the dimension [m]. This motivates the following definition of a quantity $\Phi \equiv kL$

$$\Phi \;\equiv\; kL \;\equiv\; \frac{3}{16}\int_{-\infty}^{\infty}\sum_{i=1}^{d} R_{ii}(r_y\vec{e}_2,\vec{x},t)\mathrm{d}r_y \tag{B.4}$$

which can be interpreted as a product of k and integral length scale L. According to Rotta [23] the factor 3/16 is used in order to match the value of L with a similarly

defined integral length scale denoted L_g for the case of isotropic turbulence.
From the Navier-Stokes equations for $\vec{u}$ and the derived equation for $\vec{u}'$, the following exact equation for Φ for the shear flow (B.1) can be derived, see Rotta [23] and also Menter and Egorov [18]

$$\begin{aligned}
&\frac{\partial \Phi}{\partial t} \; + \; \vec{U}\cdot\vec{\nabla}\Phi - \nu\Delta\Phi \; + \; \frac{3}{16}\int_{-\infty}^{\infty}\left[\frac{\partial U(\vec{x}+r_y\vec{e}_2)}{\partial y} - \frac{\partial U(\vec{x})}{\partial y}\right]\sum_{i=1}^{d} R_{ii}(\vec{x})\mathrm{d}r_y \\
= \; &-\frac{3}{16}\frac{\partial U(\vec{x})}{\partial y}\int_{-\infty}^{\infty} R_{21}(\vec{x})\mathrm{d}r_y - \frac{3}{16}\int_{-\infty}^{\infty}\frac{\partial U(\vec{x}+r_y\vec{e}_2)}{\partial y}R_{12}\mathrm{d}r_y \\
&+ \; \frac{3}{8}\nu\int_{-\infty}^{\infty}\sum_{i,k=1}^{d}\frac{\partial^2 R_{ii}}{\partial x_k^2}\mathrm{d}r_y \\
&+ \; \frac{3}{16}\int_{-\infty}^{\infty}\sum_{i,k=1}^{d}\frac{\partial}{\partial x_k}\left(R_{(ik)i} - R_{i(ik)}\right)\mathrm{d}r_y - \frac{\partial}{\partial y}\frac{3}{16}\int_{-\infty}^{\infty}\sum_{i=1}^{d}\left[R_{(i2)i} + \frac{2}{\rho}\langle p'v'\rangle\right]\mathrm{d}r_y
\end{aligned}$$

where ρ denotes the density. For the present shear flow the so-called triple correlations are defined by (see [23])

$$\begin{aligned}
R_{(ik)j} &= \langle u_i(\vec{x},t)u_k(\vec{x},t)u_j(\vec{x}+r_y\vec{e}_2,t)\rangle \\
R_{i(jk)} &= \langle u_i(\vec{x},t)u_j(\vec{x}+r_y\vec{e}_2,t)u_k(\vec{x}+r_y\vec{e}_2,t)\rangle.
\end{aligned}$$

Menter and Egorov now focus on the first two terms on the right-hand side. Using the fact that $R_{12} = R_{21}$ and using a Taylor series expansion for the term

$$\frac{\partial U(\vec{x}+r_y\vec{e}_2)}{\partial y} \; = \; \frac{\partial U(\vec{x})}{\partial y} \; + \; \frac{\partial^2 U(\vec{x})}{\partial y^2}r_y \; + \; \frac{1}{2}\frac{\partial^3 U(\vec{x})}{\partial y^3}r_y^2 \; + \; \mathcal{O}(r_y^3)$$

and substitution, Menter and Egorov obtain

$$\begin{aligned}
&-\frac{3}{16}\frac{\partial U(\vec{x})}{\partial y}\int_{-\infty}^{\infty} R_{21}(\vec{x})\mathrm{d}r_y \; - \; \frac{3}{16}\int_{-\infty}^{\infty}\frac{\partial U(\vec{x}+r_y\vec{e}_2)}{\partial y}R_{12}(\vec{x})\mathrm{d}r_y \\
= \; &-\frac{3}{16}\frac{\partial U(\vec{x})}{\partial y}\int_{-\infty}^{\infty} R_{21}(\vec{x})\mathrm{d}r_y \\
&- \; \frac{3}{16}\int_{-\infty}^{\infty}\left[\frac{\partial U(\vec{x})}{\partial y} \; + \; \frac{\partial^2 U(\vec{x})}{\partial y^2}r_y \; + \; \frac{1}{2}\frac{\partial^3 U(\vec{x})}{\partial y^3}r_y^2 \; + \; \mathcal{O}(r_y^3)\right]R_{12}(\vec{x})\mathrm{d}r_y \\
= \; &-\frac{6}{16}\frac{\partial U(\vec{x})}{\partial y}\int_{-\infty}^{\infty} R_{12}(\vec{x})\mathrm{d}r_y \\
&- \; \frac{\partial^2 U(\vec{x})}{\partial y^2}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x})r_y\mathrm{d}r_y \; - \; \frac{1}{2}\frac{\partial^3 U(\vec{x})}{\partial y^3}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x})r_y^2\mathrm{d}r_y \; - \; \ldots
\end{aligned}$$

The first term in the last equation is modeled using the relation:

$$\frac{6}{16}\frac{\partial U(\vec{x})}{\partial y}\int_{-\infty}^{\infty} R_{12}(\vec{x})\mathrm{d}r_y \; \approx \; \zeta_1 L\langle u'v'\rangle\frac{\partial U(\vec{x})}{\partial y}\,, \quad \zeta_1 \in \mathbb{R}. \tag{B.5}$$

The crucial point is the treatment of the remaining terms. Rotta only considered isotropic turbulence, where the two-point correlation R_{12} is symmetric with respect to $r_y = 0$. Thereafter the integral of the second term which is linear in r_y vanishes. For that reason, Rotta dropped this term in his model. The third term is non-zero

even in isotropic turbulence and was modeled by Rotta using an approximation similar to (B.5) as

$$\frac{1}{2}\frac{\partial^3 U}{\partial y^3}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x}) r_y^2 \mathrm{d}r_y \;=\; \zeta_3 L^3 \langle u'v' \rangle \frac{\partial^3 U}{\partial y^3} \;, \quad \zeta_3 \in \mathbb{R}.$$

Menter and Egorov argued that the integral of $\int R_{12} r_y \mathrm{d}r_y$ is non-zero in non-homogeneous flows, which are of general interest. Menter and Egorov propose the following modelling for this term

$$\frac{\partial^2 U}{\partial y^2}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x}) r_y \mathrm{d}r_y \;=\; \zeta_2 L^2 \langle u'v' \rangle \frac{\partial^2 U}{\partial y^2} \;, \quad \zeta_2 \in \mathbb{R}.$$

Using the eddy viscosity hypothesis for $\langle u'v' \rangle$, i.e., $\langle u'v' \rangle = -\nu_t \partial U/\partial y$, this term can be written as

$$\begin{aligned}
\frac{\partial^2 U}{\partial y^2}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x}) r_y \mathrm{d}r_y \;&= \zeta_2 L^2 \; \langle u'v' \rangle \frac{\partial^2 U}{\partial y^2} \\
&= \; -\zeta_2 L^2 \; \nu_t \frac{\partial U}{\partial y} \; \frac{\partial^2 U}{\partial y^2} \\
&= \; -\zeta_2 L^2 \; \nu_t \left(\frac{\partial U}{\partial y}\right)^2 \; \frac{\frac{\partial^2 U}{\partial y^2}}{\frac{\partial U}{\partial y}}.
\end{aligned}$$

In the last step this term reads

$$\frac{\partial^2 U}{\partial y^2}\frac{3}{16}\int_{-\infty}^{\infty} R_{12}(\vec{x}) r_y \mathrm{d}r_y \;=\; -\frac{\zeta_2}{\kappa} L \mathcal{P} \; \frac{L}{L_{\mathrm{vK}}} \;, \quad \kappa = 0.41 \tag{B.6}$$

with so-called production term $\mathcal{P}$ and von Kármán length scale L_{vK} given by

$$\mathcal{P} \;=\; \nu_t \left(\frac{\partial U}{\partial y}\right)^2 \tag{B.7}$$

$$L_{\mathrm{vK}} \;=\; \kappa \frac{\left|\frac{\partial U}{\partial y}\right|}{\left|\frac{\partial^2 U}{\partial y^2}\right|} \;, \quad \kappa = 0.41. \tag{B.8}$$

The motivation of the Kármán-sensor is based on [5].

Bibliography

[1] G.D. Birkhoff. *Proof of the ergodic theorem.* Proceedings of the National Academy of Sciences USA, 17, 656-660. 1931.

[2] F.K. Chow, P. Moin. *A further study of numerical errors in large-eddy simulations.* J. Comput. Phys. 184, 366-380. 2003.

[3] W. Dahmen, A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler.* Springer-Verlag. 2005.

[4] L. Davidson. *Evaluation of the SST-SAS Model: Channel Flow, Asymmetric Diffuser and Axi-Symmetric Hill.* European Conference on Computational Fluid Dynamics. 2006.

[5] Deutsches Zentrum für Luft-und Raumfahrt. *Technical Report – Technical Documentation of the DLR TAU-Code.* Institute of Aerodynamics and Flow Technology Braunschweig. 2009.

[6] D.M. Driver, H.L. Seegmiller. *Features of reattaching turbulent shear layer in divergent channel flow.* AIAA J. 23, 163-171. 1985.

[7] S. Faure, J. Laminie, R. Temam. *Colocated Finite Volume Scheme for Fluid Flows.* Commun. Comput. Phys., Vol.4, No.1, pp.1-25. 2008.

[8] J.H. Ferziger, M. Perić. *Computational Methods for Fluid Dynamics.* Springer-Verlag, 3.Auflage. 2002.

[9] A. Haselbacher, J. Blazek. *On the Accurate and Efficient Discretisation of the Navier-Stokes Equations on Mixed Grids.* AIAA J. 99-33552. 1998.

[10] V. John. *Large Eddy Simulation of Turbulent Incompressible Flows. Analytical and Numerical Results for a Class of LES Models.* Lecture Notes in Computational Science and Engineering 34, Springer-Verlag. 2004.

[11] P. Knabner, L. Angermann. *Numerik partieller Differentialgleichungen.* Springer-Verlag. 2000.

[12] T. Knopp, X. Zhang, R. Kessler, G. Lube. *Enhancement of an industrial finite-volume code for large-eddy-type simulation of incompressible high Reynolds number flow using near-wall modeling.* Comput. Methods Appl. Mech. Engrg.. 2008.

[13] T. Knopp, X. Zhang, R. Kessler, G. Lube. *Calibration of a Finite Volume Discretisation and of Model Parameters for Incompressible Large Eddy-Type Simulation.* Preprint, Elsevier Science. 2008.

[14] A.W. Kyle, D.L. Bonhaus. *An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids.* Computers Fluids Vol. 23, No. 1, pp 1-21. 1994.

[19] H. Le, P. Moin, J. Kim. *Direct numerical simulation of turbulent flow over a backward-facing step.* J. Fluid Mech. vol. 330, 349-374. 1997.

[16] G. Lube. *Theorie und Numerik Elliptischer Randwertprobleme (Sommersemester 2006).* Lecture Notes p.11. 2006.

[17] A. Meister. *Numerik linearer Gleichungssysteme.* Vieweg Verlag, Wiesbaden, 2.Auflage. 2005.

[18] F.R. Menter, Y. Egorov. *Revisiting the Turbulent Scale-Equation.* ANSYS. 2006.

[19] P. Moin, K. Mahesh. *Direct Numerical Simulation: A Tool in turbulent research.* Annu. Rev. Fluid Mech. 30, 539-578. 1998.

[20] T.D. Moser, J. Kim, N.N. Mansour. *Direct numerical simulation of turbulent channel flow up to* $Re_\tau = 590$. Phys. Fluids 13, 11, 943-946. 1999.

[21] S.B. Pope. *Turbulent Flows.* Cambridge University Press. 2000.

[22] C.M. Rhie, W.L. Chow. *Numerical study of turbulent flow past an airfoil with trailing edge separation.* AIAA J. 21, 1525-1532. 1983.

[23] J.C. Rotta. *Turbulente Strömungen.* B.G. Teubner Stuttgart, p.179. 1972.

[24] J. Smagorinsky. *General circulation experiments with the primitive equations, I. The basic experiment.* Mon. Weather Rev. 91, 99-152. 1963.

[25] P. Sonneveld. *CGS: A fast Lanczos-Type Solver for Nonsymmetric Linear Systems.* SIAM J. Sci. Stat. Comput., 10 (1): 36-52. 1989.

[26] P.R. Spalart. *Direct numerical simulation of a turbulent boundary layer up to* $R_\theta = 1410$. J. Fluid Mech. 187, 61-98. 1988.

[27] J.H. Spurk, N. Aksel. *Strömungslehre.* Springer-Verlag, 6. Auflage. 2005.

[28] A.A. Townsend. *The structure of turbulent shear flow.* Cambridge University Press. 1976.

[29] H. A. van der Vorst. *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems.* SIAM J. Sci. Stat. Comput., 13:631-644. 1992.

[30] J. van Kan. *A second order accurate pressure-correction scheme for viscous incompressible flow.* SIAM J. Sci. Statist. Comput., 7, 870-891. 1986.

[31] Th. von Kármán. *Mechanische Ähnlichkeit und Turbulenz.* Proc. 3rd. Congress of Applied Mechanics Stockholm, pp. 85-93. 1930.

[32] X. Xiao, J.R. Edwards, H.A. Hassas. *Blending Functions in Hybrid Large-Eddy/Reynolds-Averaged Navier-Stokes Simulations*. AIAA Journal, Vol 42, pp 2508-2515. 2004.

[33] X. Zhang. *Identification of Model and Grid Parameters for Incompressible Turbulent Flows*. Dissertation der Math.-Naturwiss. Fakultäten der Georg-August-Universität zu Göttingen, p.32. 2007.